LEANDRO WASKO
LUCIANA GRIGÓRIO DA SILVA

SISTEMA DE CONTROLE DE GUIAS MÉDICAS BASEADAS NO PADRÃO TISS

CURITIBA
2008

LEANDRO WASKO
LUCIANA GRIGÓRIO DA SILVA

ESCOLA TÉCNICA DA UFPR
BIBLIOTECA

SISTEMA DE CONTROLE DE GUIAS MÉDICAS BASEADAS NO PADRÃO TISS

Trabalho apresentado à disciplina de Trabalho de conclusão de curso, do curso de Tecnologia em Informática da Universidade Federal do Paraná.
Professor orientador: Mário de Paula Soares

MG
004.9
W321

CURITIBA
2008

Dedicamos esse trabalho primeiramente a DEUS, e às nossas famílias pela paciência e apoio durante todo o curso.

AGRADECIMENTOS

A todos que nos deram apoio durante todo o período dedicado para a elaboração desse trabalho.

Ao professor Mário de Paula Soares, pela orientação e paciência demonstrada para conosco.

A aqueles que assim como nós, acreditaram em nossa capacidade.

A nova fonte de poder não é o dinheiro nas mãos de muitos, mas a informação nas mãos de poucos.

John Naisbitt

RESUMO

O desenvolvimento deste trabalho visa demonstrar e implantar de forma simples e de fácil entendimento, a estrutura do padrão TISS (Troca de informações em saúde suplementar) definido pela ANS (Agência Nacional de Saúde), baseada no padrão XML (*Extensible Markup Language*). Esse desenvolvimento foi realizado de modo a atender às necessidades de prestadores de serviço na área médica, com a intenção de facilitar o preenchimento das guias médicas para posterior envio, e reduzir o tempo utilizado para a realização dessa atividade por parte dos usuários.

A utilização de sistemas informatizados para a geração das guias médicas possibilita uma maior confiabilidade nos dados emitidos, visto que a possibilidade de erros diminuiu gradativamente.

Palavras-chave: envio, guias médicas, tecnologia da informação, TISS.

ABSTRACT

The development of this work aims to demonstrate and deploy a simple and easy to understand the structure of the standard Tiss (exchange of information on health supplement) defined by the ANS (National Health Agency), based on standard XML (Extensible Markup Language). This development was carried out in order to meet the needs of service providers in the medical area, with the intention to facilitate the completion of medical guides for later transmission, and reduce the time taken for carrying out such activity on the part of users. The use of computerized systems for the generation of guides medical enables greater reliability in the data issued since the possibility of errors decreased gradually.

SUMÁRIO

1. **INTRODUÇÃO** 12
2. **OBJETIVOS** 14
3. **PLANO DE PROJETO** 15
4. **JUSTIFICATIVA** 17
5. **PADRÃO TISS** 21
5.1 GUIAS 21
5.1.1 GUIA DE CONSULTA 22
5.1.2 GUIA SP/SADT 22
5.1.3 GUIA DE SOLICITAÇÃO DE INTERNAÇÃO 22
5.1.4 GUIA RESUMO DE INTERNAÇÃO 22
5.1.5 GUIA HONORÁRIO INDIVIDUAL 23
5.1.6 GUIA OUTRAS DESPESAS 23
5.1.7 GUIA DE TRATAMENTO ODONTOLÓGICO - SOLICITAÇÃO 23
5.1.8 GUIA DE TRATAMENTO ODONTOLÓGICO - COBRANÇA 23
5.1.9 DEMONSTRATIVO DE RETORNO 24
5.1.10 DEMONSTRATIVO DE PAGAMENTO 24
5.1.11 DEMONSTRATIVO DE ANÁLISE DE CONTA MÉDICA 24
5.1.12 GUIA DE TRATAMENTO ODONTOLÓGICO – DEMONSTRATIVO DE PAGAMENTO 25
5.2 MENSAGENS ELETRÔNICAS 25
5.3 PADRÃO DE COMUNICAÇÃO 26
5.3.1 PREMISSAS 26
5.3.1.1 PADRÕES DE NOTAÇÃO 27
5.3.1.2 VERSIONAMENTO E DOWNLOAD 28
5.3.1.3 FORMAS DE COMUNICAÇÃO 28
5.4 PRAZOS PARA AS ADEQUAÇÕES AO TISS 29
6. **TECNOLOGIAS UTILIZADAS** 30
6.1 JSP 30
6.2 JAVA 30
6.3 XML 32
6.3.1 CARACTERÍSTICAS DO XML 33
6.3.1.1 REPRESENTAÇÃO ESTRUTURADA DOS DADOS 33
6.3.1.1.1 XSD 34
6.3.1.2 SEPARAÇÃO ENTRE DADOS E APRESENTAÇÃO 36
6.3.1.3 PRINCIPAIS BENEFÍCIOS DA LINGUAGEM XML 37
6.4 BANCO DE DADOS MYSQL 37
6.4.1 CARACTERÍSTICAS 38
6.5 AMBIENTE DE DESENVOLVIMENTO IDE ECLIPSE 38
6.6 SERVIDOR WEB APACHE TOMCAT 39
7 SIMPLE TYPE E COMPLEX TYPES 40
7.1 SIMPLE TYPE 40
7.2 COMPLEX TYPE 40
8 API XERCES 43
9 HASH 45
9.1 HASH MD5 45
10 DIAGRAMA CASOS DE USO 47
11 DESCRIÇÃO RESUMIDA DOS CASOS DE USO 48
12 LOGAR SISTEMA 50

12.1 Evento do caso de uso 50
12.2 Diagrama de telas 50
12.3 Diagrama de seqüência 51
12.4 Diagrama de comunicação 51
12.5 Diagrama de atividades 52
13 CADASTRAR USUÁRIOS 53
13.1 Evento do caso de uso 53
13.2 Diagrama de telas – Cadastrar 54
13.3 Diagrama de seqüência – Cadastrar 55
13.4 Diagrama de comunicação – Cadastrar 55
13.5 Diagrama de atividades – Cadastrar 56
13.6 Diagrama de telas – Pesquisar 57
13.7 Diagrama de seqüência – Pesquisar 57
13.8 Diagrama de comunicação – Pesquisar 58
13.9 Diagrama de atividades – Pesquisar 58
13.10 Diagrama de telas – Alterar 59
13.11 Diagrama de seqüência – Alterar 60
13.12 Diagrama de comunicação – Alterar 60
13.13 Diagrama de atividades – Alterar 61
13.14 Diagrama de telas – Excluir 62
13.15 Diagrama de seqüência – Excluir 63
13.16 Diagrama de comunicação – Excluir 63
13.17 Diagrama de atividades – Excluir 64
14. CADASTRAR CONFIGURAÇÕES 65
14.1 Evento de caso de uso 65
14.2 Diagrama de telas 65
14.3 Diagrama de seqüência 66
14.4 Diagrama de comunicação 66
14.5 Diagrama de atividades 67
15. CADASTRAR OPERADORAS 68
15.1 Evento de caso de uso 68
15.2 Diagrama de telas - Cadastrar 70
15.3 Diagrama de seqüência – Cadastrar 70
15.4 Diagrama de comunicação – Cadastrar 71
15.5 Diagrama de atividades – Cadastrar 71
15.6 Diagrama de telas – Pesquisar 72
15.7 Diagrama de seqüência – Pesquisar 72
15.8 Diagrama de comunicação – Pesquisar 73
15.9 Diagrama de atividades – Pesquisar 73
15.10 Diagrama de telas – Alterar 74
15.11 Diagrama de seqüência – Alterar 75
15.12 Diagrama de comunicação – Alterar 75
15.13 Diagrama de atividades – Alterar 76
15.14 Diagrama de telas – Excluir 77
15.15 Diagrama de seqüência – Excluir 78
15.16 Diagrama de comunicação – Excluir 78
15.17 Diagrama de atividades – Excluir 79
16. CADASTRAR BENEFICIÁRIOS 80
16.1 Evento do caso de uso 80
16.2 Diagrama de telas – Cadastrar 81

16.3 Diagrama de seqüência – Cadastrar 82
16.4 Diagrama de comunicação – Cadastrar 82
16.5 Diagrama de atividades – Cadastrar 83
16.6 Diagrama de telas – Pesquisar 84
16.7 Diagrama de seqüencia – Pesquisar 84
16.8 Diagrama de comunicação – Pesquisar 85
16.9 Diagrama de atividades – Pesquisar 85
16.10 Diagrama de telas – Alterar 86
16.11 Diagrama de seqüência – Alterar 87
16.12 Diagrama de comunicação – Alterar 87
16.13 Diagrama de atividades – Alterar 88
16.14 Diagrama de telas – Excluir 89
16.15 Diagrama de seqüência – Excluir 90
16.16 Diagrama de comunicação – Excluir 90
16.17 Diagrama de atividades - Excluir 91
17. CADASTRAR PRESTADORES 92
17.1 Eventos do caso de uso 92
17.2 Prestador físico 94
17.2.1 Diagrama de telas – Cadastrar 94
17.2.2 Diagrama de seqüência – Cadastrar 95
17.2.3 Diagrama de comunicação – Cadastrar 95
17.2.4 Diagrama de atividades – Cadastrar 96
17.2.5 Diagrama de telas – Alterar 97
17.2.6 Diagrama de seqüência – Alterar 98
17.2.7 Diagrama de comunicação – Alterar 98
17.2.8 Diagrama de atividades – Alterar 99
17.3 Prestador jurídico 100
17.3.1 Diagrama de telas – Cadastrar 100
17.3.2 Diagrama de seqüência – Cadastar 101
17.3.3 Diagrama de comunicação – Cadastrar 101
17.3.4 Diagrama de atividades – Cadastar 102
17.3.5 Diagrama de telas – Alterar 103
17.3.6 Diagrama de seqüência – Alterar 104
17.3.7 Diagrama de comunicação – Alterar 104
17.3.8 Diagrama de atividades – Alterar 105
17.4 Diagrama de telas – Pesquisar 106
17.5 Diagrama de seqüência – Pesquisar 106
17.6 Diagrama de comunicação – Pesquisar 107
17.7 Diagrama de atividades – Pesquisar 107
17.8 Diagrama de telas – Excluir 108
17.9 Diagrama de seqüência – Excluir 109
17.10 Diagrama de comunicação – Excluir 109
17.11 Diagrama de atividades - Excluir 110
18. GERAR GUIAS 111
18.1 Eventos do caso de uso 111
18.2 Diagrama de telas 112
18.3 Diagrama de seqüência 113
18.4 Diagrama de comunicação 113
18.5 Diagrama de atividades 114
19. GERAR XML DAS GUIAS 115

19.1 Eventos do caso de uso 115
19.2 Diagrama de telas 116
19.3 Diagrama de seqüência 117
19.4 Diagrama de comunicação 118
19.5 Diagrama de atividades 118
20. IMPRIMIR GUIAS 119
20.1 Eventos do caso de uso 119
20.2 Diagrama de telas 120
20.3 Diagrama de seqüência 121
20.4 Diagrama de comunicação 122
20.5 Diagrama de atividades 122
21. CONSULTAR GUIA 123
21.1 Eventos do caso de uso 123
21.2 Diagrama de telas 124
21.3 Diagrama de seqüencia 125
21.4 Diagrama de comunicação 125
21.5 Diagrama de atividades 126
22. DIAGRAMA DE CLASSES 127
23. DIAGRAMA DE OBJETOS 128
24. DER 129
25. CÓDIGO FONTE 130
26. CONCLUSÃO 499
REFERENCIAS 500
ANEXOS 502

1. INTRODUÇÃO

Em maio de 2003, tendo-se verificado a necessidade de padronização na troca de informações entre operadoras de planos de saúde e prestadores da área de saúde, a ANS iniciou pesquisas para a elaboração de uma estrutura única e obrigatória de comunicação entre as partes envolvidas.

As operadoras, na época, utilizavam grandes quantidades de formulários para receberem dos prestadores informações necessárias à área administrativa das mesmas. Esses formulários não traziam confiabilidade, por existir a possibilidade de preenchimentos incorretos, além da demora nas transações e um custo administrativo altamente elevado. Após essas análises, percebeu-se a necessidade dessa padronização.

Em fevereiro de 2005, a ANS lançou uma minuta de resolução normativa e iniciou uma consulta pública em seu site, para o levantamento de sugestões e críticas referente ao assunto.

Após essa consulta, a equipe da ANS agrupou todas as informações e sugestões à Resolução Normativa 114/2005, chegando-se à simplificação das guias e padrões com a preocupação de não se perder a qualidade das informações trocadas.

Com essa resolução normativa a ANS estabeleceu um padrão de troca de informações em saúde suplementar, conhecido como TISS. Esta padronização veio unificar a comunicação entre prestadores de serviços de saúde e operadoras de planos privados de assistência a saúde, de

maneira fácil através da utilização da linguagem XML como padrão de comunicação.

O padrão TISS tende a aproximar as operadoras e prestadores de serviços de saúde através de uma uniformização do intercâmbio de dados por um sistema eletrônico de envio de informações, onde o maior beneficiário, o consumidor de plano de saúde, passasse a receber melhoria de atendimento com agilidade.

Com essa padronização, a necessidade de criação de sistemas que informatizam os procedimentos e tarefas para o envio das guias médicas seguindo esses padrões, se faz cada vez mais necessária.

Com a assimilação do conhecimento adquirido durante o período do curso, e pesquisas referentes às tecnologias utilizadas foi desenvolvido um sistema web que proporciona ao usuário maior facilidade no preenchimento dessas guias médicas, e disponibiliza os dados informados em formato XML, seguindo a estrutura estabelecida pela ANS.

2. OBJETIVOS

1. OBJETIVO GERAL

Criar um aplicativo informatizado para a área de saúde, que visa atender prestadores de serviços que não tenham sistemas próprios para se adequarem ao padrão de comunicação da Troca de Informações em Saúde Suplementar (TISS). Este aplicativo deve operar em ambiente web, permitindo o cadastro de usuários prestadores, beneficiários e operadoras, bem como gerar dinamicamente guias médicas padronizadas pela Agência Nacional de Saúde (ANS) em tela, permitindo o preenchimento das mesmas pelo usuário e geração dos dados informados em formato XML com estrutura padrão estabelecidos pela ANS, possibilitando a comunicação correta com as operadoras.

2. OBJETIVOS ESPECÍFICOS

- Permitir aos usuários o gerenciamento cadastral de operadoras de plano de saúde, prestadores de serviços e beneficiários.
- Permitir aos usuários a geração, em tempo de execução, das guias médicas.
- Permitir a impressão das guias seguindo a padronização TISS.
- Garantir a integridade das informações geradas no arquivo XML, utilizando o hash.
- Permitir a consulta e visualização das guias médicas por data e tipo de guias emitidas.

3. PLANO DE PROJETO

Escopo do projeto

O sistema deverá permitir em tempo de execução a geração de guias médicas do padrão TISS, possibilitando assim o preenchimento dos campos das guias e após isso a geração em arquivo XML dos dados informados, que também deve ser dinâmico.

Essas guias após serem geradas em formato XML, serão armazenadas possibilitando consultas em determinado período escolhido pelo usuário.

Outra característica que o sistema deve possuir é possibilitar a impressão das guias no formato visual estabelecido pela ANS.

Processo de desenvolvimento

Desenvolvimento de código fonte:

- Recursos humanos: com conhecimentos em Java, JSP e XML, dois recursos.
- Ferramentas de software: IDE Eclipse 3.4, Banco de dados MySql 5.0 , Servidor Web TomCat 6.04
- Hardware: 1 computador Athlon XP 1.7, 1 GB de RAM e HD 160 GB e 1 computador Intel Core 2 6320 1.41 mhz, 1 GB de RAM e HD 160 GB;
- Procedimentos adotados: pesquisas sobre as tecnologias utilizadas foram realizadas durante o período aproximado de dois meses. Os desenvolvedores fizeram em conjunto a construção da prototipação inicial do projeto. Todos desenvolveram por se

tratar de uma linguagem onde não possuíamos conhecimento suficiente.

Documentação

- Recursos humanos: necessário conhecimento em modelagem UML, utilizados dois recursos.
- Ferramentas de software: utilizado o JUDE como ferramenta de diagramação.
- Hardware: 1 computador Athlon XP 1.7, 1 GB de RAM e HD 160 GB e 1 computador Intel Core 2 6320 1.41 mhz, 1 GB de RAM e HD 160 GB.
- Procedimentos adotados: a documentação foi elaborada a medida que o desenvolvimento do projeto foi ocorrendo. Foi validado parte da documentação junto ao orientador, em períodos irregulares.

Equipe de projeto

Leandro Wasko – desenvolvimento e documentação

Luciana Grigorio da Silva – desenvolvimento e documentação

4. JUSTIFICATIVA

De acordo com o ANS (Agência Nacional de Saúde Suplementar), a confiabilidade e segurança da informação são categorias que preocupam os mais variados setores da atualidade. Na área de saúde isso não é diferente. A privacidade de informações individuais, entre as quais se incluem aquelas referentes ao estado de saúde de cada beneficiário, é ponto focal quando se fala em transferência eletrônica de dados.

Uma mensagem ou transação eletrônica é um conjunto estruturado de informações trocado entre atores de diversos setores com a finalidade de solicitar uma operação ou informar um resultado.

A implantação de um sistema informatizado, que possibilitasse a continuidade da privacidade e veracidade das informações enviadas, foi amplamente discutida e analisada de modo a garantir não somente uma padronização, mas também uma facilidade para a administração e gerenciamento no fluxo do processo.

Dentro dessa realidade o objetivo do padrão TISS é o de atingir a compatibilidade e interoperabilidade entre os diversos sistemas independentes e gerar redução substancial na manipulação de formulários e no tempo de processamento das informações.

Após o período de pesquisas, a ANS optou por adotar o XML/Schema (*Extensible MarkUp Language*) como linguagem padrão para troca de mensagens eletrônicas, por se tratar de um padrão flexível, portátil e reconhecido internacionalmente para compatibilidade entre sistemas e dados. A proposta da ANS utiliza padrões já existentes e disponíveis em outros

bancos de dados e sistemas de informação da Agência e do Ministério da Saúde, tais com SIB, SIP, SINASC, SINAM, CIH.

O padrão TISS (Troca de Informação em Saúde Suplementar), estabelecido pela Agência Nacional de Saúde (ANS), surgiu com o objetivo principal de padronizar as transações eletrônicas de informações administrativas e financeiras, de modo a aprimorar a eficiência e efetividade do sistema de saúde.

> "No Brasil hoje são mais de 40 milhões de usuários cobertos por planos de saúde sendo o setor o segundo maior em termos de gastos no mundo. Estamos atrasados na questão de TI e sistemas informatizados no campo de saúde. A padronização será uma revolução". (Jussara Rötzsch, gerente geral de integração com o SUS).

Devido ao grande número de formulários existentes na época, não se tinha um padrão de guias, ou seja, cada operadora utilizava seu próprio modelo e isso dificultava o trabalho dos prestadores, pois além de terem que realizar o preenchimento manualmente, precisavam saber a diferenciação de guia para cada operadora. Isso exigia uma demanda de tempo altamente elevada e intensa dedicação burocrática dos prestadores de serviços na área de saúde.

Como se a burocratização já não fosse problema suficiente, a lentidão dos processos administrativos no atendimento dos beneficiários, assim como as possibilidades de erros e as dificuldades de comparação de dados, eram conseqüências notoriamente associadas ao excesso de papéis e à ausência de sistemas de informação unificados e ágeis.

Com a implantação do TISS, os prestadores de serviço puderam submeter à mesma guia/transação para qualquer operadora, o que diminuiu o índice de perdas financeiras e tempo gasto. A adoção de padrões eletrônicos

simplificou significativamente os processos, o que ajudou na redução de custos administrativos.

A Resolução Normativa nº 114/2005 determinou a adoção de forma gradual e obrigatória ao padrão TISS por parte das operadoras de planos privados de assistência à saúde e dos prestadores de serviço em todo Brasil.

> "O desenvolvimento e implementação de aplicativo informatizado para troca destas informações, poderá facilitar ou mesmo permitir, que pequenas e médias operadoras e prestadores de serviços de saúde participem desta troca de maneira eletrônica, ganhando assim agilidade, diminuição de custos e melhores condições de competir no mercado", diz Jussara Rötzsch

Dentre os diversos benefícios trazidos pelo TISS destacam-se:

- O aprimoramento da comunicação entre os prestadores de serviços e operadoras de planos de saúde privados.

- A redução do uso de papel e quantidade de formulários para preenchimento;

- Redução de custos administrativos e erros de preenchimento;

- Melhoria na qualidade da assistência à saúde;

Conforme publicado na Instrução Normativa nº 22 de 16 de novembro de 2006, o layout dos formulários deve ser mantido integralmente, não sendo permitida nenhuma alteração em sua estrutura.

RADAR TISS

O Radar Tiss foi definido na resolução normativa 153 de 28 de maio de 2007, e tem como objetivo realizar estudos sobre a implantação de padrão Tiss.

A primeira coleta de dados realizada no período entre 20 de novembro de 2007 e 20 de janeiro de 2008 apresentou resultados satisfatórios sobre o tema.

De acordo com a pesquisa, em outubro de 2007, o total de guias trocadas entre prestadores e operadoras chegou perto de 26 milhões, o que demonstra uma aceitação favorável ao sistema implantado.

O estudo identificou como os diferentes segmentos de prestadores de serviços e operadoras em todas as regiões do Brasil estão se adaptando à padronização da informação e às guias eletrônicas.

De acordo com a pesquisa, a região sul foi a que apresentou o maior número de trocas eletrônicas no período observado, chegando a 60,2% em comparação com as demais regiões.

Outro aspecto importante observado após a pesquisa refere-se aos campos obrigatórios exigidos por cada operadora, cerca de (85%) desses campos não são exigidos por menos da metade das operadoras, o que contribuiu para diferenciações entre um mesmo tipo de guia.

BIBLIOTECA DA ESCOLA TECNICA DA UFRJ

5. PADRÃO TISS

O manual do padrão TISS, encontrado no site da ANS, define que as categorias de padrões na área de informática em saúde são de comunicação, de vocabulário, de conteúdo e estrutura, de privacidade, confidencialidade e segurança. De forma a garantir que esses padrões sejam seguidos, o TISS definiu o seguinte:

- Padrão de comunicação: a linguagem de marcação XML/Schema;
- Padrão de vocabulário: o CID 10, por exemplo, é um dos padrões adotados na área públicos e privados para a descrição dos diagnósticos do paciente;
- Padrão de conteúdo e estrutura: são os padrões definidos nas guias e demonstrativos;
- Padrão de privacidade, confidencialidade e segurança: foram adotadas as normas editadas pelo Conselho Federal de Medicina.

O padrão TISS é composto por guias e demonstrativos de pagamento, mensagens eletrônicas e estrutura SCHEMA-XML.

5.1 GUIAS

As guias do padrão TISS são modelos formais e obrigatórios de representação de descrição documental sobre os eventos assistenciais realizados no beneficiário e enviadas do prestador para a operadora.

O layout dos formulários deve ser seguido integralmente, conforme publicado na Instrução Normativa nº 22 de 16 de novembro de 2006, não sendo possível nenhum tipo de alteração em sua estrutura.

Nos anexos desse documento, encontram alguns exemplos das guias aqui descritas.

5.1.1 GUIA DE CONSULTA

A guia de consulta é utilizada somente na execução de consultas eletivas, onde não existam procedimentos e constitui-se no documento padrão para a solicitação de pagamento.

5.1.2 GUIA SP/SADT

A Guia de Serviços Profissionais / Serviço Auxiliar Diagnóstico e Terapia (SP/SADT), utilizada no atendimento a diversos tipos de eventos: remoção, pequena cirurgia, terapias, consulta com procedimentos, exames, etc.

Compreende os processos de autorização, desde necessária, e de execução dos serviços. A consulta de referência deve ser preenchida na própria Guia de SP/SADT.

5.1.3 GUIA DE SOLICITAÇÃO DE INTERNAÇÃO

A Guia de Solicitação de Internação é o formulário padrão a ser utilizado para a solicitação, autorização ou negativa, de internação, em regime hospitalar, hospital-dia ou domiciliar.

5.1.4 GUIA RESUMO DE INTERNAÇÃO

A Guia de Resumo de Internação é o formulário padrão a ser utilizado para a finalização do faturamento da internação.

5.1.5 GUIA HONORÁRIO INDIVIDUAL

A Guia de Honorário Individual é utilizada para a apresentação do faturamento de honorários profissionais prestados em serviços de internação, caso estes sejam pagos diretamente ao profissional.

5.1.6 GUIA OUTRAS DESPESAS

A Guia de Outras Despesas é utilizada nos casos de apresentação do faturamento em papel, como instrumento de continuidade e complemento de folhas. Esta guia estará sempre ligada a uma guia principal (Guia de SP/SADT ou Guia de Resumo de Internação), não existindo por si só.

Utilizada para discriminação de materiais, medicamentos, aluguéis, gases e taxas diversas não informadas na guia principal.

5.1.7 GUIA DE TRATAMENTO ODONTOLÓGICO - SOLICITAÇÃO

A Guia Odontológica / Solicitação tem como finalidade ser utilizada para descrever a situação inicial do paciente e para solicitação dos procedimentos odontológicos por cirurgiões-dentistas, tanto em consultórios quanto em clínicas odontológicas.

5.1.8 GUIA DE TRATAMENTO ODONTOLÓGICO - COBRANÇA

A Guia Odontológica / Cobrança tem como finalidade ser utilizada para faturamento dos procedimentos odontológicos realizados por cirurgiões-dentistas, tanto em consultórios quanto em clínicas odontológicas.

Qualquer tipo de procedimento odontológico executado no paciente é registrado na Guia Odontológica/Cobrança.

5.1.9 DEMONSTRATIVO DE RETORNO

São modelos formais de representação e descrição documental do padrão TISS, sobre o pagamento dos eventos assistenciais e enviados da operadora para o prestador.

O Demonstrativo de Pagamento e o Demonstrativo de Análise de Conta Médica são os documentos pelos quais as operadoras enviarão aos prestadores as informações relativas ao faturamento e processamento das Guias.

5.1.10 DEMONSTRATIVO DE PAGAMENTO

Os Demonstrativos de Pagamento são documentos enviados da operadora para o prestador com a finalidade de fornecer extrato das contas da produção apresentadas nas guias em questão e seu pagamento ou não.

5.1.11 DEMONSTRATIVO DE ANÁLISE DE CONTA MÉDICA

Os Demonstrativos de Análise de Conta Médica são documentos enviados da operadora para o prestador com a finalidade de fornecer informações detalhadas sobre o processamento do lote de guias de faturamento enviado pelo prestador, item a item. A partir deste demonstrativo é possível ao prestador fazer uma previsão das contas a serem pagas pela operadora e solicitar revisão de possíveis glosas baseado nos detalhes do processamento das guias.

5.1.12 GUIA DE TRATAMENTO ODONTOLÓGICO – DEMONSTRATIVO DE PAGAMENTO

Os Demonstrativos de Pagamento são documentos enviados da operadora para o prestador com a finalidade de fornecer extrato das contas da produção apresentadas nas guias em questão e seu pagamento ou não. Neste caso, esta guia tem o objetivo de apresentar a prestação de contas referente às guias odontológicas apresentadas pelo prestador de serviço.

5.2 MENSAGENS ELETRÔNICAS

Buscando a adoção de um padrão flexível e reconhecido internacionalmente, a ANS, com a contribuição das operadoras e dos prestadores de serviços, optou por adotar o XML (extensible MarkUp Language) como linguagem para troca de mensagens eletrônicas.

O padrão XML está definido em http://www.w3.org/XML/ (W3C-World Wide Web Consortium).

Para validação da estrutura de uma mensagem XML, são necessários os arquivos Schema-XML Schema Definition (XSD).

A Resolução Normativa definiu a estrutura lógica das mensagens e a Instrução Normativa nº 26, de 10 de maio de 2007, publicou a versão 2.01.02 dos seus formatos físicos.

A mensagem TISS, é composta por: Cabeçalho, corpo da mensagem e epílogo, que contém o código *hash*.

O cabeçalho de uma mensagem TISS é composto por: Identificação da transação + seqüencial da transação + data + hora + origem + destino.

A estrutura do corpo da mensagem TISS padroniza o conteúdo de informação que deve ser transacionado entre prestadores e operadoras de planos de saúde.

O epílogo da mensagem TISS contém o *hash* das informações. Esse campo contém o *hash* apenas dos valores contidos nas transações, não incluindo as tags XML. O objetivo do *hash* é garantir que o dado, uma vez gravado, não seja alterado.

5.3 PADRÃO DE COMUNICAÇÃO

O padrão de comunicação TISS é composto de quatro schemas XML:

- Schema com tipos de dados simples TissSimpleType;
- Schema com tipo de dados complexos, construídos a partir dos tipos de dados simples, TissComplexTypes;
- Schema com a descrição das Guias que compõem o padrão TISS TissGuias;
- Schema que descreve a Mensagem TISS TissV2_01_03.xsd, este schema define as trocas entre prestadores e operadoras, utilizando os elementos definidos nos três schemas anteriores.

Os schemas são constantemente atualizados, sendo que atualmente se encontram na versão 03.

5.3.1 PREMISSAS

Na construção do schema que define o padrão TISS, algumas premissas foram adotadas:

- Adoção dos padrões já utilizados pela ANS, Ministério da Saúde e das sugestões da consulta pública nº 21/2005 sobre o padrão TISS;

- Inclusão do maior número possível de domínios no schema-XML para aumentar a capacidade de expressão do modelo;
- Incorporação de diferentes alternativas de fornecimento da mesma informação para aumentar a flexibilidade;
- Construção de tipos de dados simples e, a partir deles, criação de elementos de maior complexidade que correspondem à agregação dos tipos simples para, finalmente, se definir a Mensagem TISS;

5.3.1.1 PADRÕES DE NOTAÇÃO

- O namespace "ans:" foi criado para identificar os elementos do schema do padrão TISS;
- Os elementos que descrevem tipos simples de dados foram identificados com o prefixo "st_". Exemplo: st_tipoAtendimento, st_tipoConsulta. Tipos simples que foram utilizados em várias guias estão descritos no schema TissSimpleTypesV2_01_03.xsd, encontrado no site do TISS na seção download;
- Os elementos que descrevem tipos complexos de dados foram identificados com o prefixo "ct_". Os tipos complexos foram construídos a partir dos tipos simples. Tipos complexos que foram utilizados em várias guias estão descritos no schema TissComplexTypesV2_01_03.xsd, encontrado no site do TISS na seção download;
- O nome dos elementos foi definido por extenso para manter a expressividade do padrão e facilitar o seu entendimento.

5.3.1.2 VERSIONAMENTO E DOWNLOAD

Os schemas que compõem o padrão TISS são versionados. O número da versão é identificado pelo nome do arquivo que contém o schema. Acrescentou-se ao nome do arquivo que contém o schema o ano (AAAA), o mês (MM) e número da versão do padrão exemplo, TissSimpleTypesV2_01_03.xsd.

Os schemas iniciaram com a versão V2_01_01, como sofrem atualizações constantemente, a versão dos schemas é a V2_01_03.

5.3.1.3 FORMAS DE COMUNICAÇÃO

As operadoras devem estar preparadas para receber de seus prestadores credenciados, os arquivos XML através de upload em sua página web, independente de outras formas de comunicação disponibilizadas opcionalmente.

Nenhum tipo de tecnologia está proibido pela Norma. Qualquer tecnologia legada poderá ser utilizada desde que consiga atender na íntegra a norma e o padrão de comunicação e segurança determinado, e não-proprietária, a fim de facilitar a utilização da mesma por todos os atores envolvidos.

As informações padronizadas devem ser aplicadas no fluxo entre Prestadores e Operadoras. Não há necessidade de padronização entre Contratada e contratante.

Todos os campos obrigatórios definidos para a guia devem estar contemplados nas interfaces, obedecendo à nomenclatura e a seqüência estabelecida no padrão.

5.4 PRAZOS PARA AS ADEQUAÇÕES AO TISS

Para que a adequação ao padrão TISS, ocorresse de forma gradativa a ANS definiu os seguintes prazos máximos para os grupos:

- Operadoras e prestadores do grupo 1: até 31 de maio de 2007;
- Prestadores do grupo 2: até 30 de novembro de 2008;
- Prestadores do grupo 3: até 30 de novembro de 2008.

6. TECNOLOGIAS UTILIZADAS

6.1 JSP

JSP (Java Server Pages) é uma tecnologia utilizada no desenvolvimento de aplicações Web, desenvolvida pela Sun microsystem, similar às tecnologias Active Server Pages (ASP) da Microsoft e PHP. Estas tecnologias estão fundamentadas na arquitetura SSI – Server Side Includes que são comandos extensivos à linguagem HTML, os quais podem conter conteúdo estático (HTML) e dinâmico (ASP, PHP, JSP, etc.). Esses comandos dinâmicos são processados pelo servidor Web antes da página HTML ser enviada. No lugar do comando é enviado apenas o resultado deste comando no formato HTML.

Por ser baseada na linguagem de programação Java, tem a vantagem da portabilidade de plataforma, que permite a sua execução em diversos sistemas operacionais, como o Windows e o Linux. Esta tecnologia permite ao desenvolvedor web produzir aplicações que acessem o banco de dados, manipulem arquivos no formato texto, capturem informações a partir de formulários e captem informações sobre o visitante e sobre o servidor.

Uma página criada com a tecnologia JSP, depois de instalada em um servidor de aplicação compatível com a tecnologia Java EE, é transformada em um Servlet.

6.2 JAVA

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90, por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens

convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "bytecode" que é executado por uma máquina virtual.

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

- Orientação a objeto - Baseado no modelo de Smalltalk e Simula67;
- Portabilidade - Independência de plataforma - "write once run anywhere";
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- Segurança - Pode executar programas via rede com restrições de execução;

MÁQUINA VIRTUAL JAVA (JVM)

Programas Java não são traduzidos para a linguagem de máquina como outras linguagens estaticamente compiladas e sim para uma representação intermediária, chamada de bytecodes.

Os bytecodes são interpretados pela máquina virtual Java (JVM - Java Virtual Machine). A JVM é responsável pelo gerenciamento dos aplicativos, à medida que são executados.

Graças à máquina virtual Java, os programas escritos em Java podem funcionar em qualquer plataforma de hardware e software que possua uma versão da JVM, tornando assim essas aplicações independentes da plataforma onde funcionam.

Mecanismos mais elaborados como o HotSpot da Sun, que guarda informações disponíveis somente em tempo de execução (ex.: número de usuários, processamento usado, memória disponível), para otimizar o

funcionamento da JVM, possibilitando que a JVM vá "aprendendo" e melhorando seu desempenho.

Os bytecodes produzidos pelos compiladores Java podem ser usados num processo de engenharia reversa para a recuperação do programa-fonte original. Esta é uma característica que atinge em menor grau todas as linguagens compiladas.

6.3 XML

Extensible Markup Language (XML) é uma linguagem de marcação de dados (*meta-markup language*) que provê um formato para descrever dados estruturados.

Isso facilita declarações mais precisas do conteúdo e resultados mais significativos de busca através de múltiplas plataformas. A maior parte da usabilidade do XML é a transferência de dados.

O XML permite a definição de um número infinito de tags. Enquanto no HTML, as tags são usadas para definir a formatação de caracteres e parágrafos, o XML provê um sistema para criar tags para dados estruturados.

Por se tratar de um documento de texto puro, ou seja, independe de qualquer plataforma, ele é muito utilizado para transferência de dados entre diferentes aplicações e sistemas.

Um elemento XML pode ter dados declarados como sendo preços de venda, taxas de preço, um título de livro, a quantidade de chuva, ou qualquer outro tipo de elemento de dado. Uma vez que o dado foi encontrado, ele pode ser distribuído pela rede e apresentado em um browser como o Internet

Explorer 5 de várias formas possíveis, ou então esse dado pode ser transferido para outras aplicações para processamento futuro e visualização.

6.3.1 CARACTERÍSTICAS DO XML

6.3.1.1 REPRESENTAÇÃO ESTRUTURADA DOS DADOS

O XML provê uma representação estruturada dos dados que mostrou ser amplamente implementável e fácil de ser desenvolvida.

Essa estrutura pode ser definida pelo próprio usuário, ou ser definida pelos *schemas*.

Dentre os padrões de definição de estrutura de um arquivo XML estão:

- DTD – Document Type Definition
- XSD – XML Schema Definition
- XDR – XML Data Reducted

Quando um documento é bem formado, ele pode ser validado utilizando a linguagem *XML Schema Definition*.

O padrão XSD (*Xml Schema Definition*) ou XS (*Xml Schema*) é a recomendação oficial do W3C desde 2001 para validação. Esse padrão fornece diversas funcionalidades, sendo possível construir tipos próprios derivados dos tipos básicos, realizar relacionamentos entre elementos de dados dentro do XML (similar aos relacionamentos entre tabelas), etc.

A ANS definiu o XSD, como schema de estruturação de dados para o Tiss, visando estabelecer um padrão de documento a ser validado.

6.3.1.1.1 XSD

Em meados de 1999, o consórcio W3C, responsável por diversos padrões da WEB, publicou pela primeira vez o XML Schema. Essa foi a primeira iniciativa de apresentar uma alternativa ao padrão DTD que era o amplamente utilizado para validações de documentos XML.

A principal motivação em propor um novo padrão era para suprir algumas das limitações que as DTDs apresentavam. Após diversas revisões, em 2001, foi lançada a recomendação final do XML Schema e em 2004 houve algumas adaptações. Hoje existem diversas APIs para validação de documentos XML, bancos de dados que dão suporte a tipos de dados XML (e o XML Schema), além de bancos de dados nativos que trabalham apenas com XML e tem o XML Schema como seu mecanismo básico de controle de consistência.

Por ser baseado em linguagem XML, o *XML Schema* obedece às mesmas regras que um documento XML. A diferença é que o XSD possui algumas necessidades que o XML não precisa. Para atender essas necessidades, é preciso definir as partes do documento XSD.

O XML permite que o usuário crie suas próprias *tags* possibilitando a flexibilidade dos documentos, isso pode acarretar em problemas para a comunicação, pois a mesma *tag* por ser definida por usuários diferentes, pode também ter significado diferente.

Para resolver esse inconveniente, o *XML Schema* trabalha com o conceito de namespace, que é um container para os nomes utilizados. Funciona como um agrupador e geralmente tem seu nome associado a um nome único (comumente uma URL).

Além de trabalhar com os namespace, o XSD também utiliza:

1. Declarações de elementos

São definidos utilizando-se a *tag* <element> a qual possui alguns atributos, os principais são:

- Name: que especifica o nome do elemento;
- Type: que especifica o tipo de dados do elemento;
- MinOccurs: define a quantidade mínima que o elemento deve aparecer no documento;
- MaxOccurs: define a quantidade máxima que o elemento pode aparecer no documento.
- Unbounded: utilizada para especificar que um elemento pode aparecer em quantidade ilimitada no documento.

2. Declaração de atributos

São definidos utilizando-se a *tag* <attribute>, e possui atributos como:

- Name: especifica o nome do atributo;
- Type: especifica o tipo de dados do atributo;
- Use: define a utilização do atributo (opcional, requerido, proibido).

3. Declaração de tipos dados

O XSD possui vários tipos de dados, além de propiciar também a criação tipos próprios. Alguns desses tipos são:

- Xsd:string: string de caracteres de tamanho ilimitado;
- Xsd:integer: número inteiro

- Xsd:date: data

6.3.1.2 SEPARAÇÃO ENTRE DADOS E APRESENTAÇÃO

A mais importante característica do XML se resume em separar a interface com o usuário (apresentação) dos dados estruturados. O HTML especifica como o documento deve ser apresentado na tela por um navegador. Já o XML define o conteúdo do documento.

O XML separa os dados da apresentação e processo, o que permite visualizar e processar o dado como quiser, utilizando diferentes folhas de estilo e aplicações. Essa separação dos dados da apresentação permite a integração dos dados de diversas fontes.

6.3.1.3 PRINCIPAIS BENEFÍCIOS DA LINGUAGEM XML

- Buscas mais eficiente;
- Desenvolvimento de aplicações flexíveis para a WEB;
- Integração de dados de fontes diferentes;
- Computações e manipulações locais;
- Múltiplas formas de visualizar os dados;
- Atualizações granulares dos documentos;
- Fácil distribuição na WEB
- Compressão.

6.4 BANCO DE DADOS MYSQL

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language* - Linguagem de Consulta Estruturada) como interface.

O servidor de banco de dados MySQL é extremamente rápido, confiável, e fácil de usar. O Servidor MySQL foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes e tem sido usado em ambientes de produção de alta demanda por diversos anos de maneira bem sucedida. Apesar de estar

em constante desenvolvimento, o Servidor MySQL oferece hoje um rico e proveitoso conjunto de funções. A conectividade, velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

6.4.1 CARACTERÍSTICAS

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem drivers ODBC, JDBC e NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP, ASP e Ruby);
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre com base na GPL;
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid;
- Suporta controle transacional;
- Interfaces gráficas ([MySQL Toolkit]) de fácil utilização cedidos pela MySQL Inc.

6.5 AMBIENTE DE DESENVOLVIMENTO IDE ECLIPSE

Eclipse é uma IDE de código aberto para a construção de programas de computador. O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade.

Hoje o Eclipse é a IDE Java mais utilizada no mundo. Possui como características marcantes o uso da SWT e não do Swing como biblioteca

gráfica, a forte orientação ao desenvolvimento baseado em plug-ins e o amplo suporte ao desenvolvedor com centenas de plug-ins que procuram atender as diferentes necessidades de diferentes programadores.

A popularidade do Eclipse se dá pelos recursos embutidos na aplicação, que facilitam sensivelmente a vida dos programadores. Além disso, o Eclipse sofre atualizações constantes, um bom nível de maturação e com versões para Windows, Linux e MacOSX.

6.6 SERVIDOR WEB APACHE TOMCAT

O Tomcat é um servidor de aplicação Java para web, mais especificamente, um . container de servlets. Desenvolvido pela Apache Software Foundation, é distribuído como software livre dentro do conceituado projeto Apache Jakarta, sendo oficialmente endossado pela Sun como a implementação de referência para as tecnologias Java Servlet e JavaServer Pages (JSP). Ele cobre parte da especificação J2EE com tecnologias como servlet e JSP, e tecnologias de apoio relacionadas como Realms e segurança, JNDI Resources e JDBC DataSources.

O Tomcat é robusto e eficiente o suficiente para ser utilizado mesmo num ambiente de produção. Como servidor web, ele provê um servidor web HTTP puramente em Java.

O servidor de aplicações responde a algumas questões comuns à todas as aplicações, como segurança, garantia de disponibilidade, balanceamento de carga e tratamento de exceções.

7 SIMPLE TYPE E COMPLEX TYPES

7.1 SIMPLE TYPE

Um *simple type* (tipo simples) restringe o valor de um atributo ou de um elemento numa instância XML. Um elemento *simpleType* é a representação de um *simple type*. Todo *simple type* é uma derivação direta ou indireta de um tipo de dado "built-in".

O seguinte *simple type* é um exemplo do padrão TISS, que define o *simple type* st_CEP.

```
<simpleType name="st_CEP">
        <restriction base="string">
                <maxLength value="8"/>
        </restriction>
</simpleType>
```

7.2 COMPLEX TYPE

Um *complex type* (tipo complexo) restringe a estrutura de um elemento numa instancia de XML. Um elemento *complexType* é a representação XML de um *complex type*. Os *complex types* fornecem as seguintes funcionalidades:

* O *complex type* possibilita adicionar tipos de atributo para *simple types*.

* O *complex type* possibilita "conteúdo vazio". Conteúdo vazio significa que uma instância XML correspondente não contém texto nem elementos aninhados.

Um complex type que especifique conteúdo vazio pode ter atributos.

* O *complex type* possibilita tipos de elementos aninhados.

* O *complex type* possibilita conteúdo misturado. Conteúdo misturado significa que uma instância XML correspondente pode conter conteúdo textual misturado com elementos.

O exemplo abaixo corresponde a um *complex type* do padrão TISS. Esse *complex type* define a estrutura beneficiário.

```
<complexType name="ct_beneficiario">

    <sequence>
        <element name="numeroCarteira" type="ans:st_descricao20"/>
        <element name="nomeBeneficiario" type="ans:st_nome"/>
        <element name="nomePlano" type="ans:st_descricao40"/>
        <element name="validadeCarteira" type="ans:st_data"
minOccurs="0"/>
        <element name="numeroCNS" type="ans:st_descricao15"
minOccurs="0"/>
        <element name="identificadorBeneficiario" type="base64Binary"
minOccurs="0"/>
    </sequence>

</complexType>
```

Nota-se no exemplo acima que, um *complex type* é constituído por *simple types*, porém o *complex type*, também pode conter outros *complex type* em sua estruturação, como no exemplo abaixo.

```
<complexType name="ct_cancelaGuia">
        <sequence>
            <element name="dadosPrestador"
type="ans:ct_contratado"/>
            <element name="guia">
                <complexType>
                    <sequence>
```

```
                                        <element name="dadosGuia"
type="ans:ct_cabecalhoGuia" maxOccurs="unbounded"/>
                                    </sequence>
                                </complexType>
                            </element>
                            <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                        </sequence>
</complexType>
```

8 API XERCES

Xerces é uma API (*Application Programming Interface*) de fonte aberta, que constitui um analisador XML de alto desempenho. Desenvolvido pela *Apache XML Project*, o Xerces traz em seu pacote os *Java Parser* DOM e SAX que são os parsers mais comuns para a implementação de interfaces XML. Além disso, o Xerces traz O *XML SCHEMA API*, que possui um módulo com um conjunto de interfaces que possibilitam o acesso a elementos do XML Schema, bem como a leitura desses arquivos.

Algumas dessas interfaces são:

- XSLoader é uma interface que define métodos para carregar documentos de XML Schema, ou seja, arquivos .xsd
- XSModel é uma interface que representa uma Schema XML. É obtido através de um XSLoader.
- XSTypeDefinition é uma interface que representa ou um tipo complexo (Complex Type) ou um tipo simples (Simple Type).
- XSComplexTypeDefinition representa uma definição de um Complex Type (o elemento complexType num documento .xsd)
- XSSimpleTypeDefinition representa uma definição de um Simple Type (o elemento simpleType num documento .xsd). A interface prove vários operações para determinar os atributos que definem o formato de um valor permitido desse tipo.
- XSParticle representa o componente Particle de uma Schema. O Particle é composto de três atributos: minOccurs, maxOccurs e o Term.

- XSTerm define um "termo" que pode ser: um model group, um corunga ou a declaração de um elemento (XSElementDeclaration, XSModelGroup ou XSWildcard)
- XSModelGroup representa o componente Model Group de um Schema. É composto por um dos tipos (All, Choice, Sequence) e de um XSObjectList.
- XSObjectList é uma abstração de uma lista ordenada, imutável de XSObject.

9 HASH

Um hash é uma seqüencia de bits geradas por um algoritmo de dispersão, em geral representada em base hexadecimal, que permite a visualização em letras (A a F) e números, representando 2 bytes cada. O conceito teórico diz que "hash é a transformação de uma grande quantidade de informações em uma pequena quantidade de informações".

Essa seqüencia busca identificar um arquivo ou informação unicamente. Por exemplo, uma mensagem de correio eletrônico, uma senha, uma chave criptográfica ou mesmo um arquivo.

É um método para transformar dados de tal forma que o resultado seja (quase) exclusivo. Além disso, funções usadas em criptografia garantem que não é possível a partir de um valor de hash retornar à informação original.

Os hashes produzem "selos de segurança" de comprimento fixo, não importa o comprimento do fluxo de dados ou do arquivo que representem. Qualquer alteração efetuada no arquivo, por mínima que seja, altera substancialmente o resultado hash. Isto ocorre porque, mesmo se apenas um dos bits do arquivo for alterado, muitos bits do resultado serão afetados. Este comportamento é conhecido como efeito avalanche.

9.1 HASH MD5

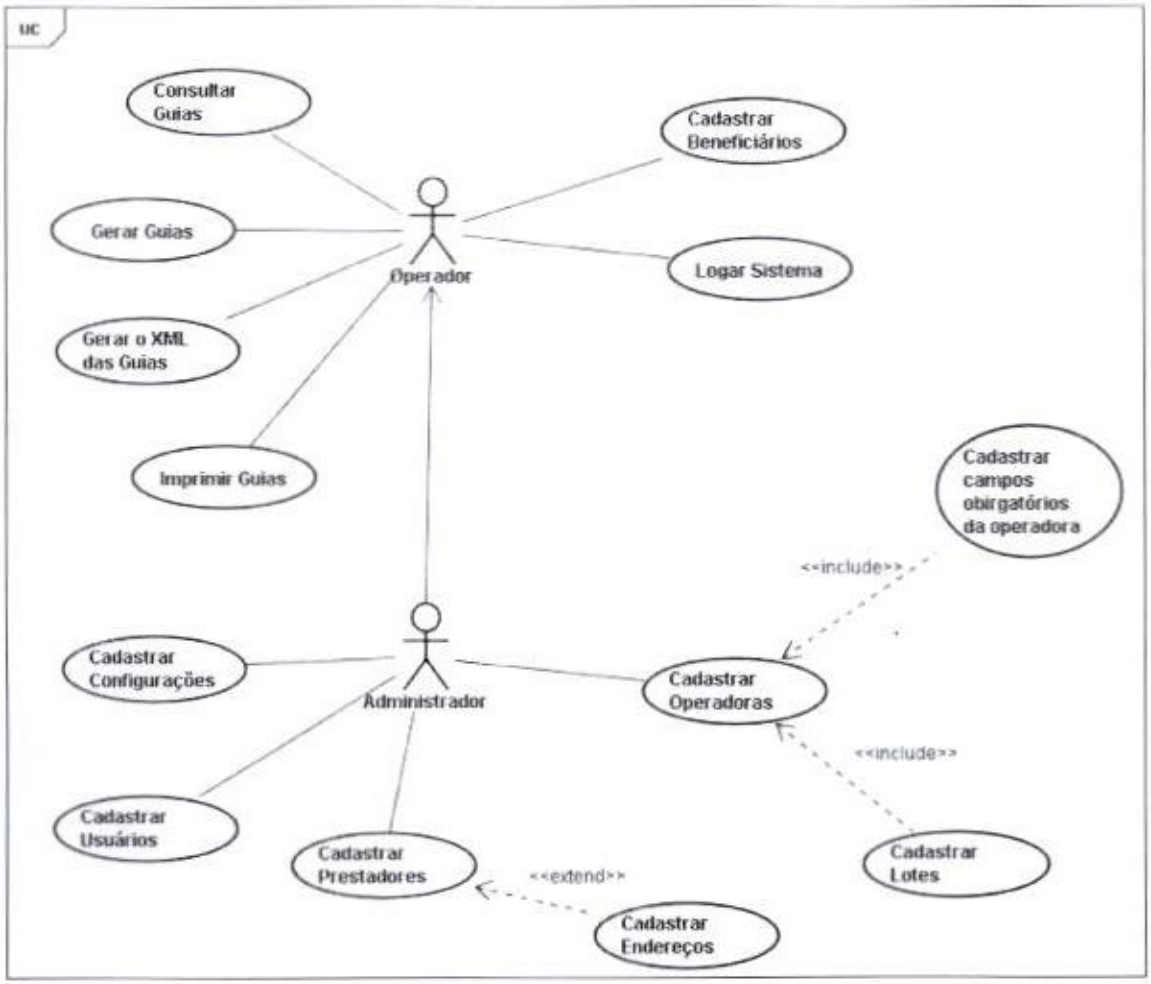
O MD5 (*Message-Digest algorithm 5*) é um algoritmo de *hash* de 128 bits unidirecional desenvolvido pela RSA Data Security, Inc., descrito na RFC 1321.

Foi desenvolvido em 1991 por Ronald Rivest para suceder ao MD4 que tinha alguns problemas de segurança. Por ser um algoritmo unidirecional, um *hash* md5 não pode ser transformado novamente no texto que lhe deu origem.

O MD5 também é usado pera verificar a integridade de um ficheiro através, por exemplo, do programa md5sum, que cria a *hash* de um ficheiro.

Como autenticação de *login* é utilizada em vários sistemas operacionais unix e em muitos *sites* com autentificação.

10 DIAGRAMA CASOS DE USO



11 DESCRIÇÃO RESUMIDA DOS CASOS DE USO

1. Logar Sistema

Este caso de uso permite o acesso do usuário ao sistema.

2. Cadastrar de usuários

Visa à inclusão de novo usuário para acesso ao sistema.

3. Cadastrar configurações

Este caso de uso visa configurar o local onde se encontra o arquivo .XSD para leitura bem como, indicar o local de armazenamento das guias geradas.

4. Cadastrar operadora

Este caso de uso visa o cadastro das diversas operadoras, para as quais serão emitidas as guias.

5. Cadastrar campos obrigatórios

Este caso de uso realiza a inclusão dos campos opcionais, que para a operadora que está sendo cadastrada, se tornam obrigatórios.

6. Cadastrar lotes

Este caso de uso visa o cadastro de lotes de guias para a emissão das mesmas.

7. Cadastrar beneficiários

Este caso realiza a inclusão dos dados dos beneficiários atendidos pelo profissional.

8. Cadastrar prestadores

É o caso de uso no qual se faz a inclusão de um novo prestador podendo ser jurídico ou físico.

9. Cadastrar endereço

Esse caso de uso refere-se ao cadastro dos endereços dos prestadores, possibilitando que o mesmo prestador tenha outros endereços no cadastro, para o caso de filiais.

10. Gerar guias

Este caso visa à geração dinâmica de cada guia para o preenchimento da mesma.

11. Gerar XML guias

Este caso de uso realiza a geração da guia em questão dentro dos padrões XML, arquivando-as numa pasta indicada.

12. Imprimir guias

Esse é o caso de uso que permite visualizar e imprimir a guia em questão.

13. Consultar guias

Esse caso uso permite a pesquisa por período das guias geradas em XML.

12 LOGAR SISTEMA

12.1 Evento do caso de uso

Pré-requisito: Estar devidamente cadastrado no sistema.

Fluxo Principal

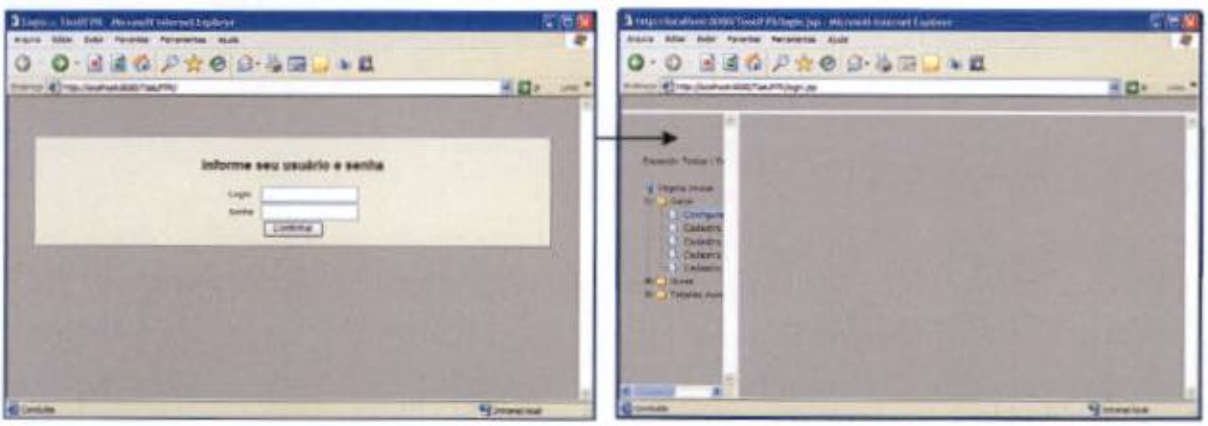
O usuário digita o endereço de acesso ao sistema no browser. A primeira tela que o sistema apresenta ao usuário é a tela de login do sistema. Nessa tela o usuário digita seu login e senha e após clica no botão confirmar.

Após verificar login, o sistema apresenta ao usuário a tela principal do sistema.

Fluxo alternativo

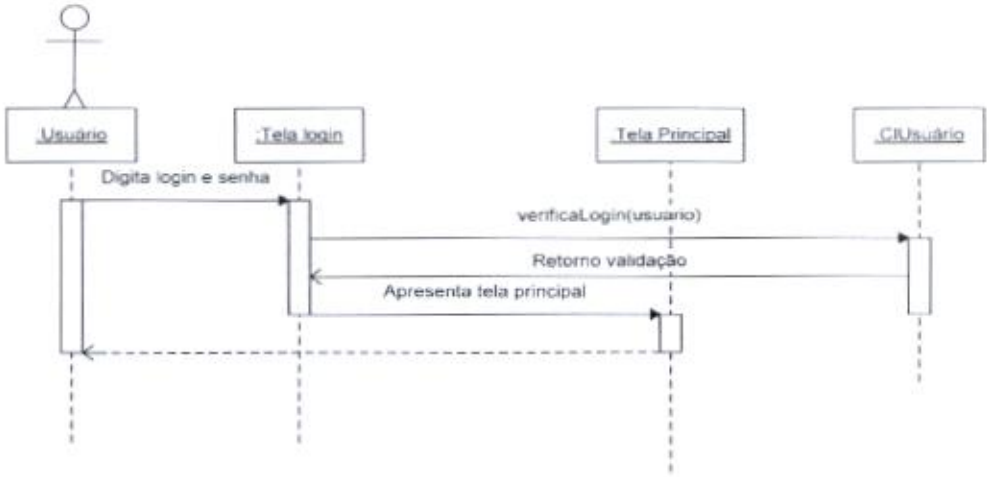
O usuário digita login e/ou senha inválidos, o sistema não permitirá o acesso e enviará uma mensagem, informando que os dados lançados não estão corretos.

12.2 Diagrama de telas



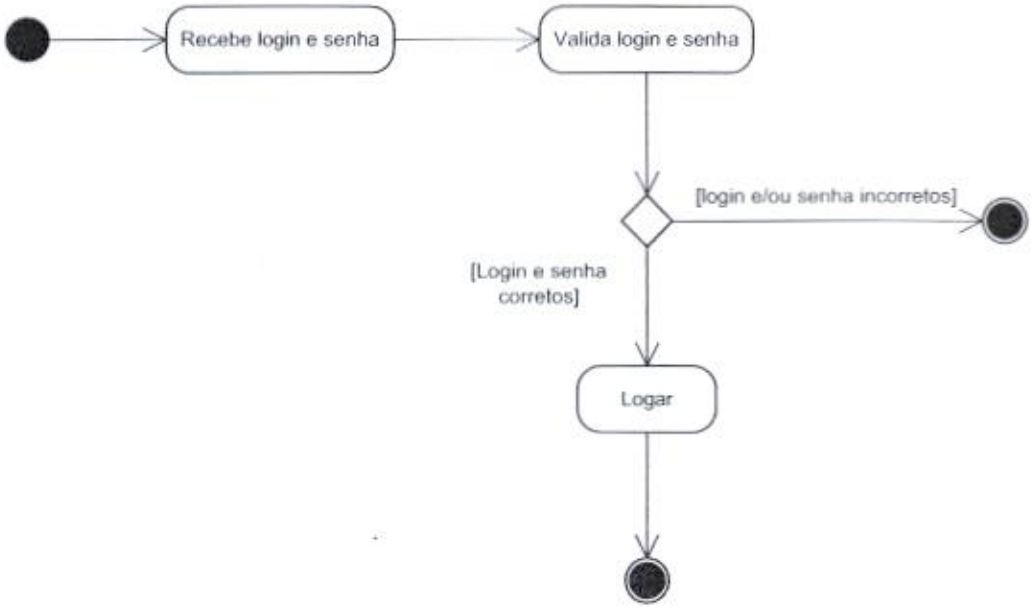
BIBLIOTECA DA ESCOLA TÉCNICA DA UFPr

12.3 Diagrama de seqüência



12.4 Diagrama de comunicação

12.5 Diagrama de atividades



13 CADASTRAR USUÁRIOS

13.1 Evento do caso de uso

Pré-requisitos: nenhum

Fluxo Principal

O administrador seleciona o link de cadastro de usuários. A tela inicial do cadastro de usuários é a de pesquisa. Nessa tela, o usuário clica no botão "Novo cadastro" que trará a tela de cadastro. Após digitar todos os dados do cadastro, o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro que acabou de ser realizado, indicando assim que foi incluso o novo cadastro.

Fluxos alternativos

Pesquisar

Na tela de pesquisa o administrador digita o nome inicial ou a letra com a qual inicia o nome a procurar, clica no botão "Pesquisar", o sistema listará todos os nomes com os parâmetros desejados.

Alterar

Na tela de pesquisa o administrador digita o nome inicial ou a letra com a qual inicia o nome a procurar, clica no botão "Pesquisar", o sistema listará todos os nomes com os parâmetros desejados. Após gerada a lista, o usuário clica no ícone alterar ( ) referente ao cadastro que deseja alterar, abrirá uma tela para a alteração dos dados. Tendo lançado todos os dados que se deseja alterar o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro que acabou de ser alterado, indicando assim que foi incluso as devidas alterações no cadastro.

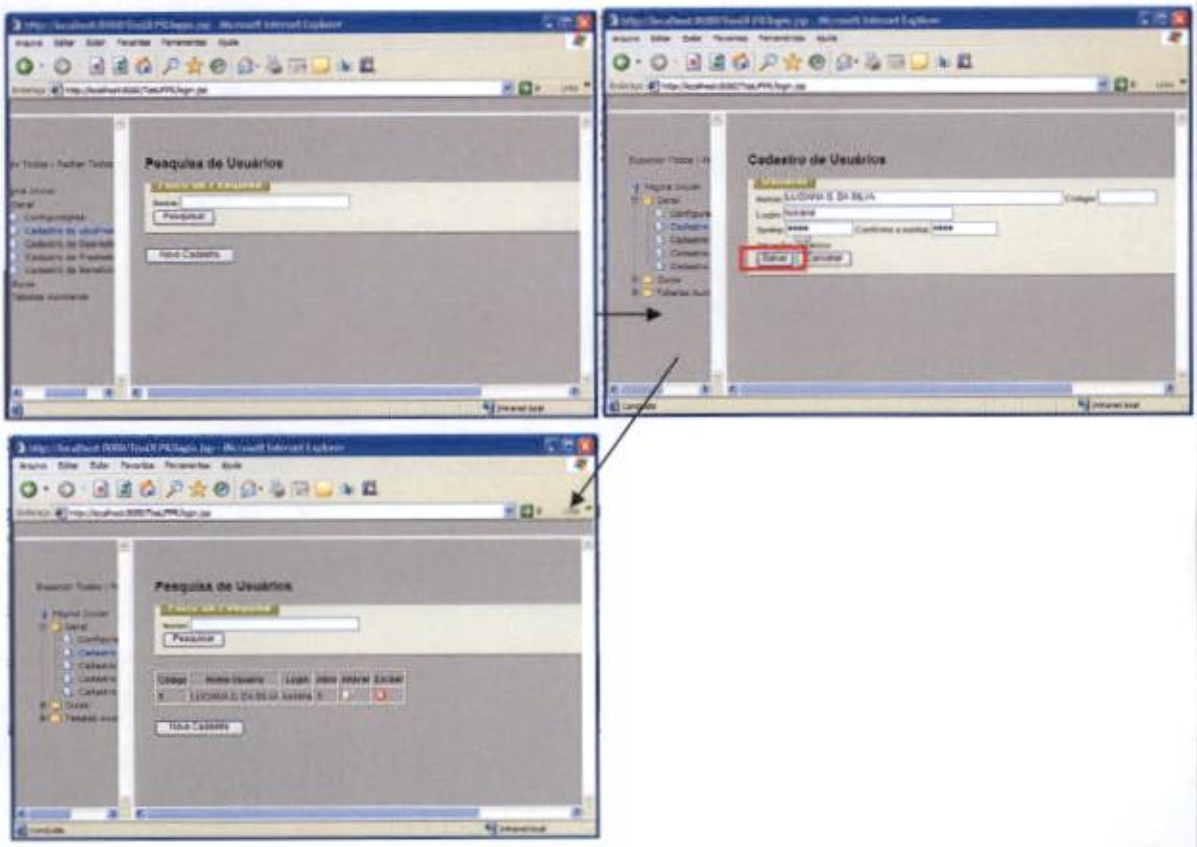
Excluir

Na tela de pesquisa o administrador digita o nome inicial ou a letra com a qual inicia o nome a procurar, clica no botão "Pesquisar", o sistema listará todos os nomes com os parâmetros desejados. Após gerada a lista, o usuário clica no ícone excluir ( ) referente ao cadastro que deseja excluir. Aparecerá um alerta solicitando a confirmação da exclusão, o usuário clicando no botão "OK", confirmando a exclusão.

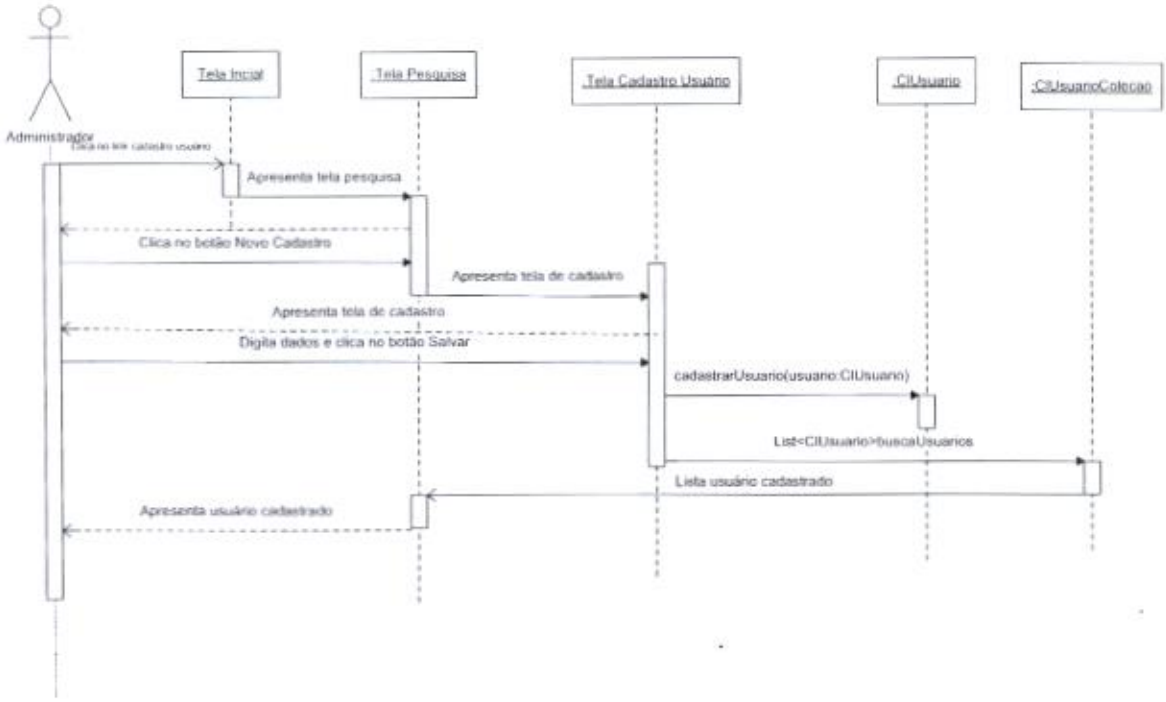
Cancelar

Se o administrador clicar em "Cancelar", o sistema interromperá o cadastro.

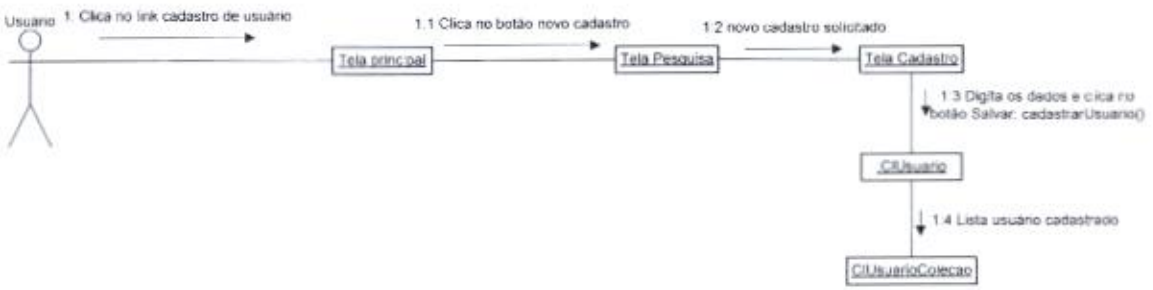
13.2 Diagrama de telas – Cadastrar



13.3 Diagrama de seqüência – Cadastrar

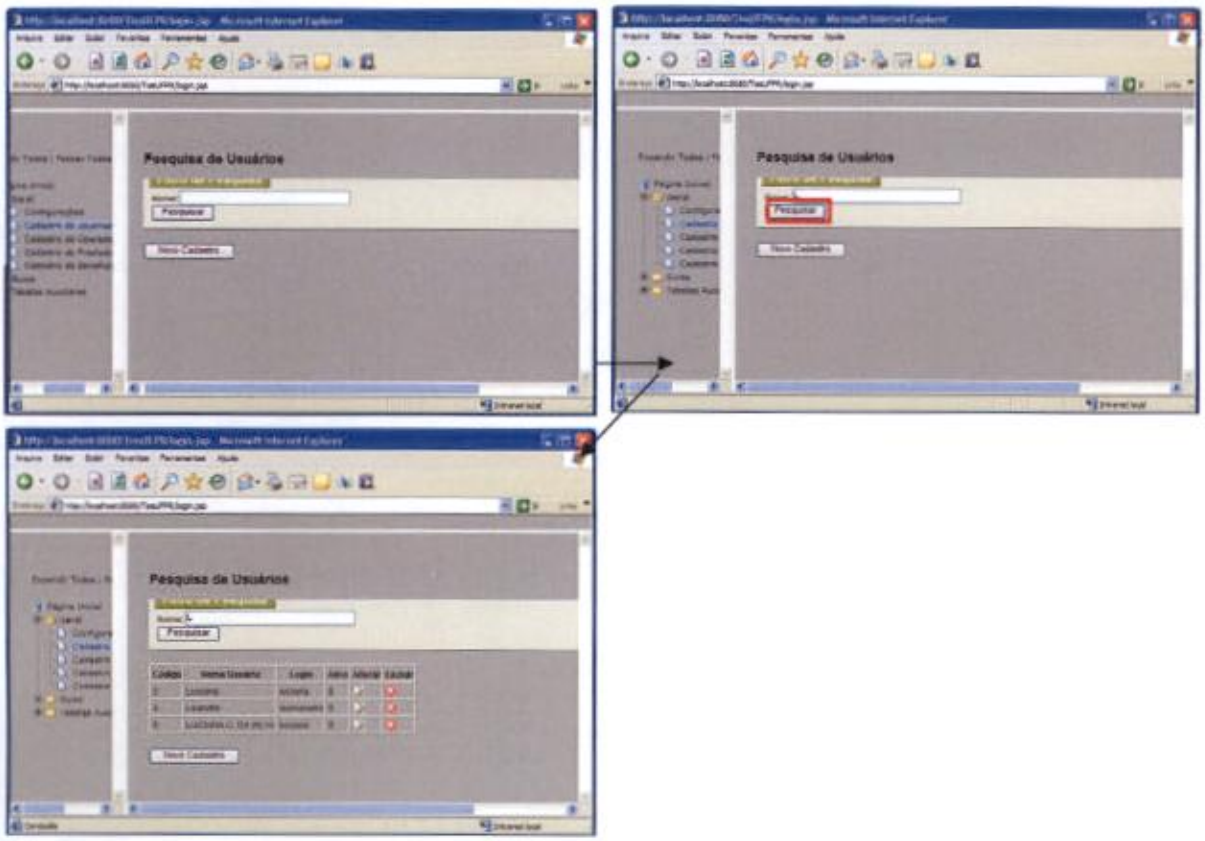


13.4 Diagrama de comunicação – Cadastrar

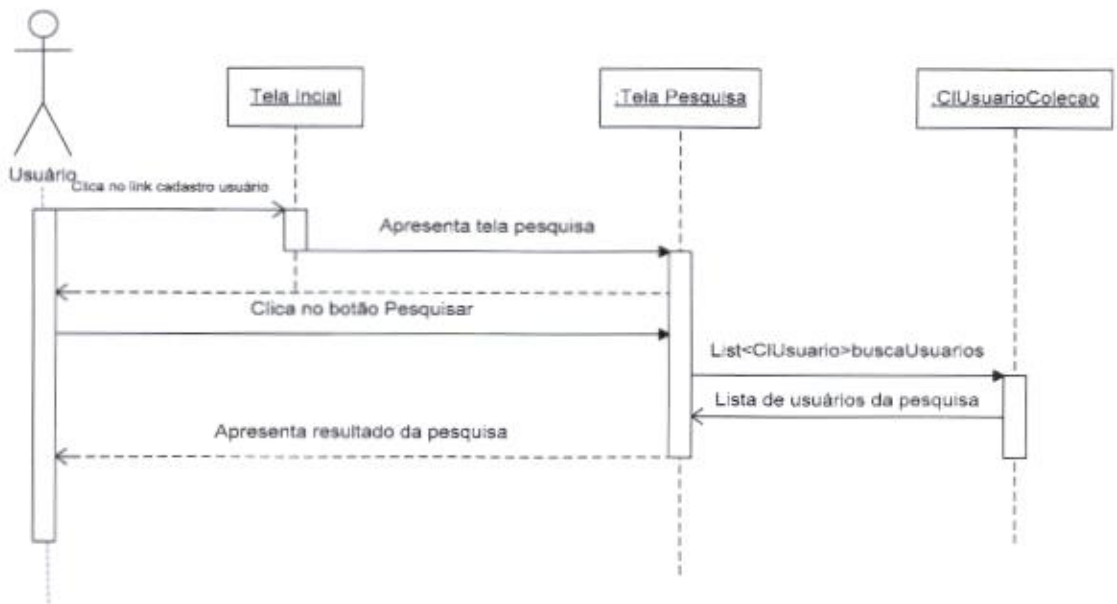


13.5 Diagrama de atividades – Cadastrar

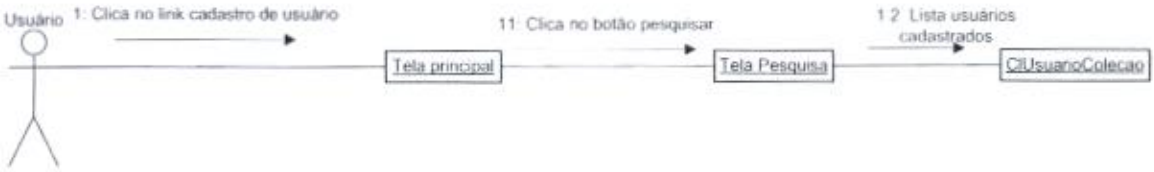
13.6 Diagrama de telas – Pesquisar



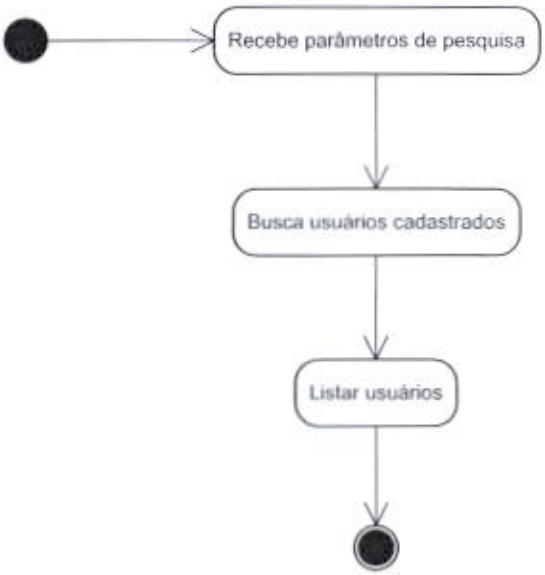
13.7 Diagrama de seqüência – Pesquisar



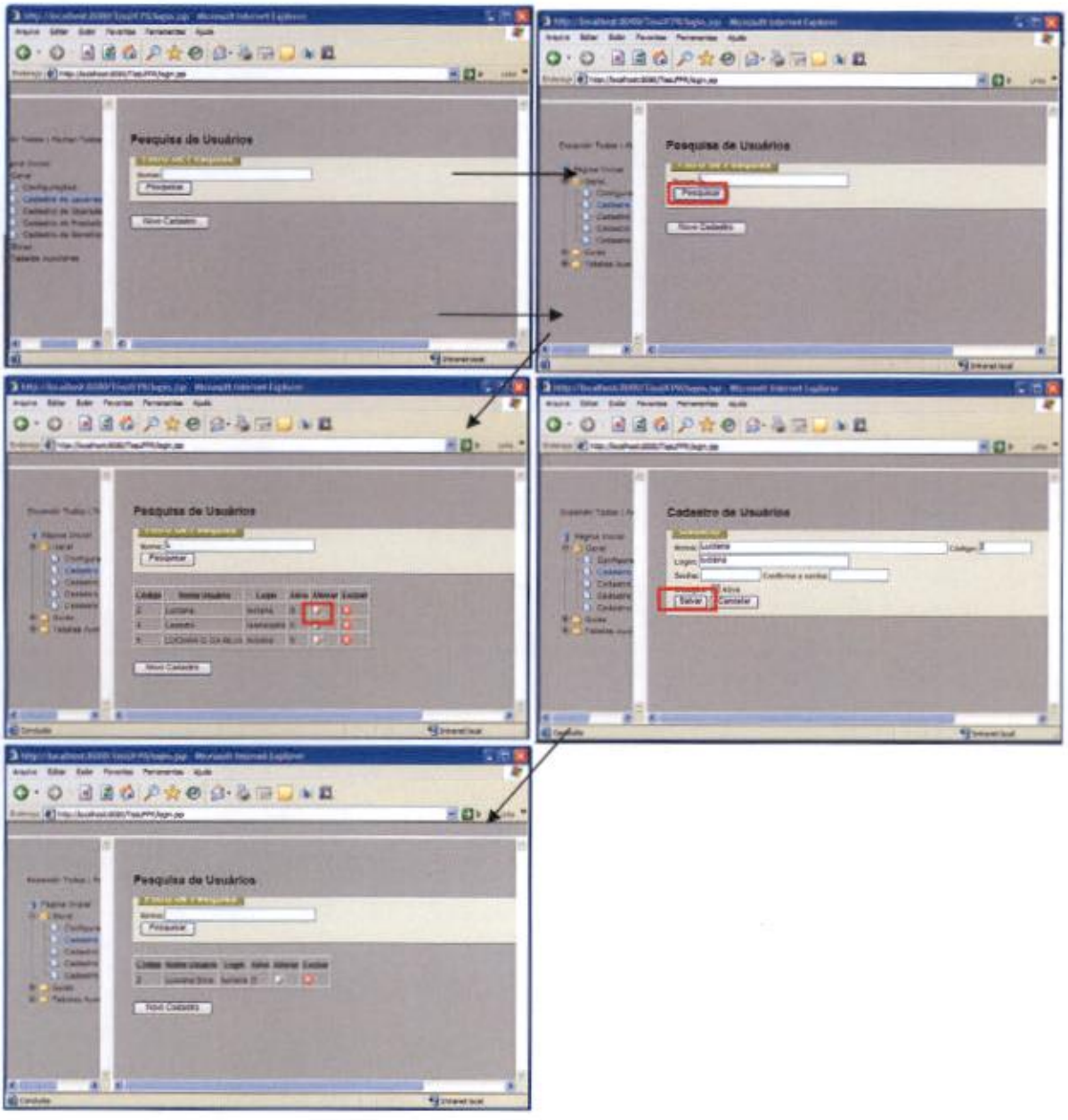
13.8 Diagrama de comunicação – Pesquisar



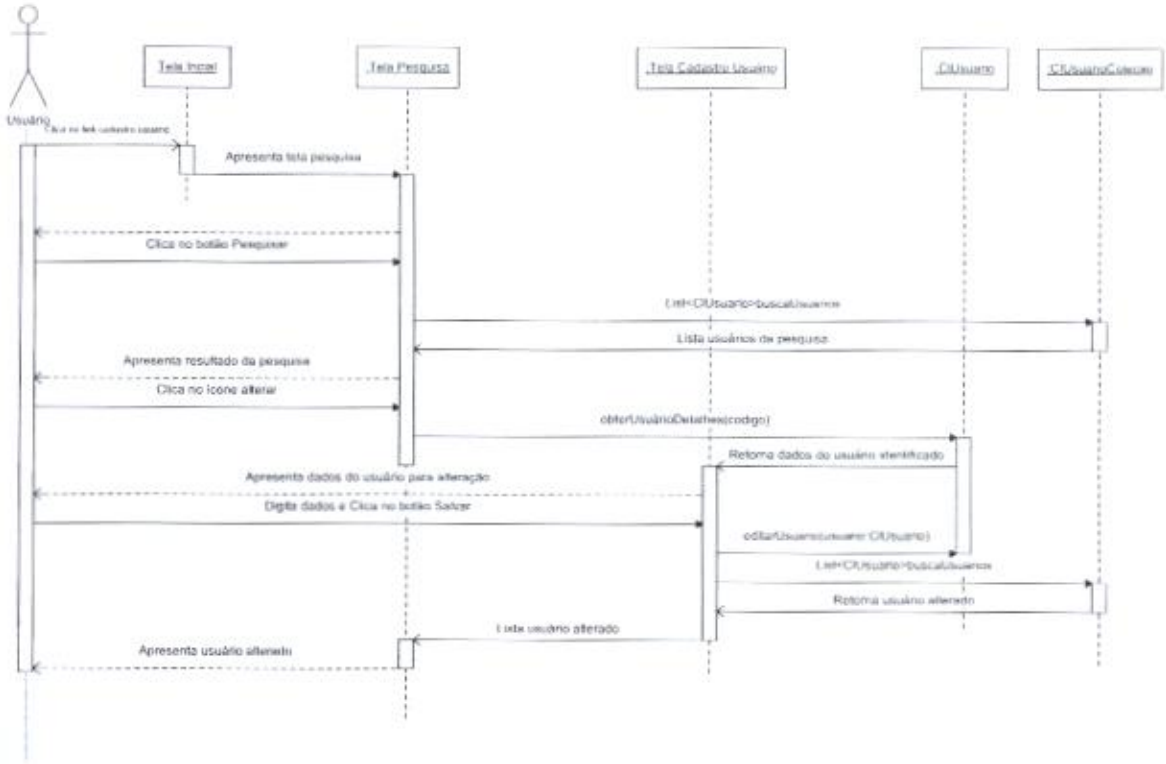
13.9 Diagrama de atividades – Pesquisar



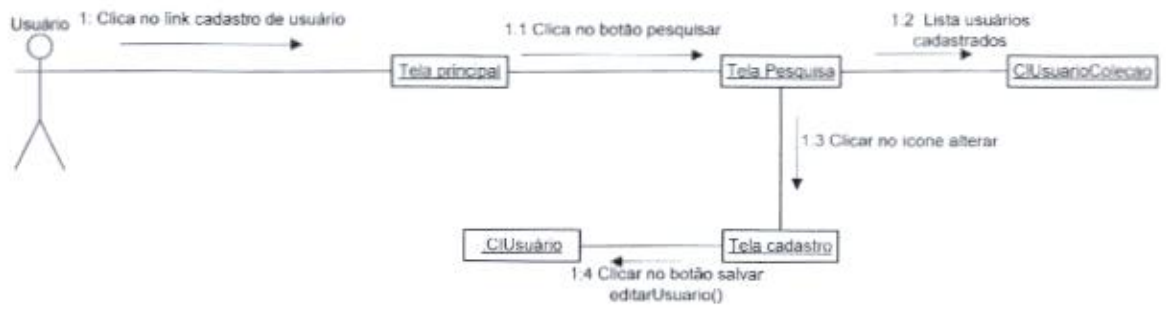
13.10 Diagrama de telas – Alterar



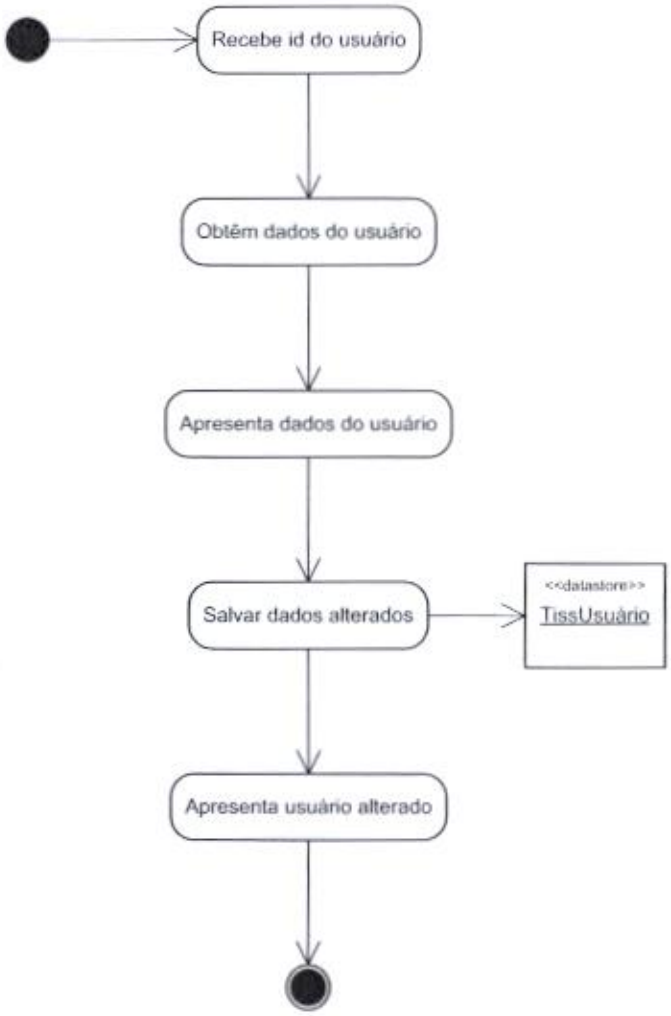
13.11 Diagrama de seqüência – Alterar



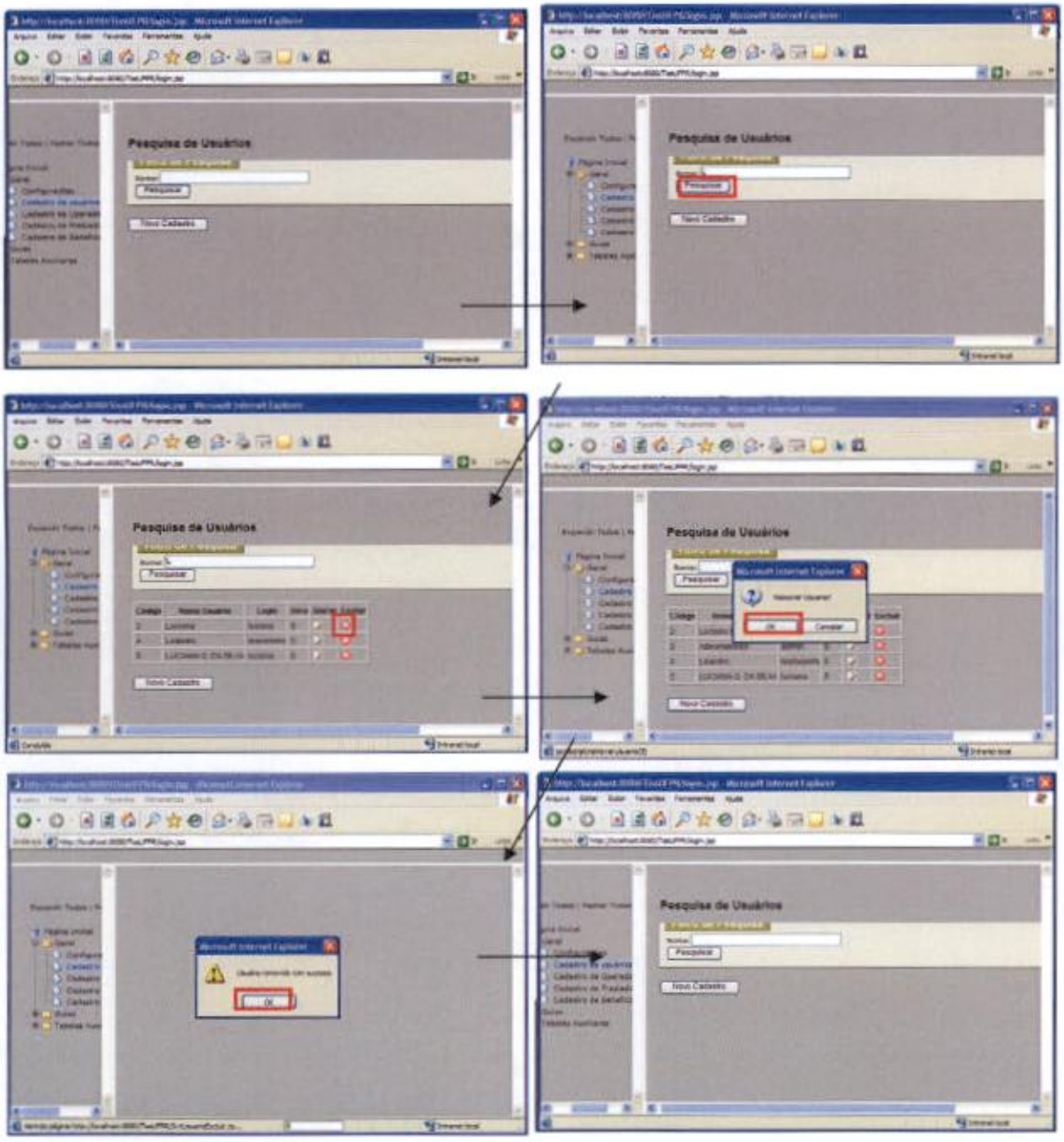
13.12 Diagrama de comunicação – Alterar



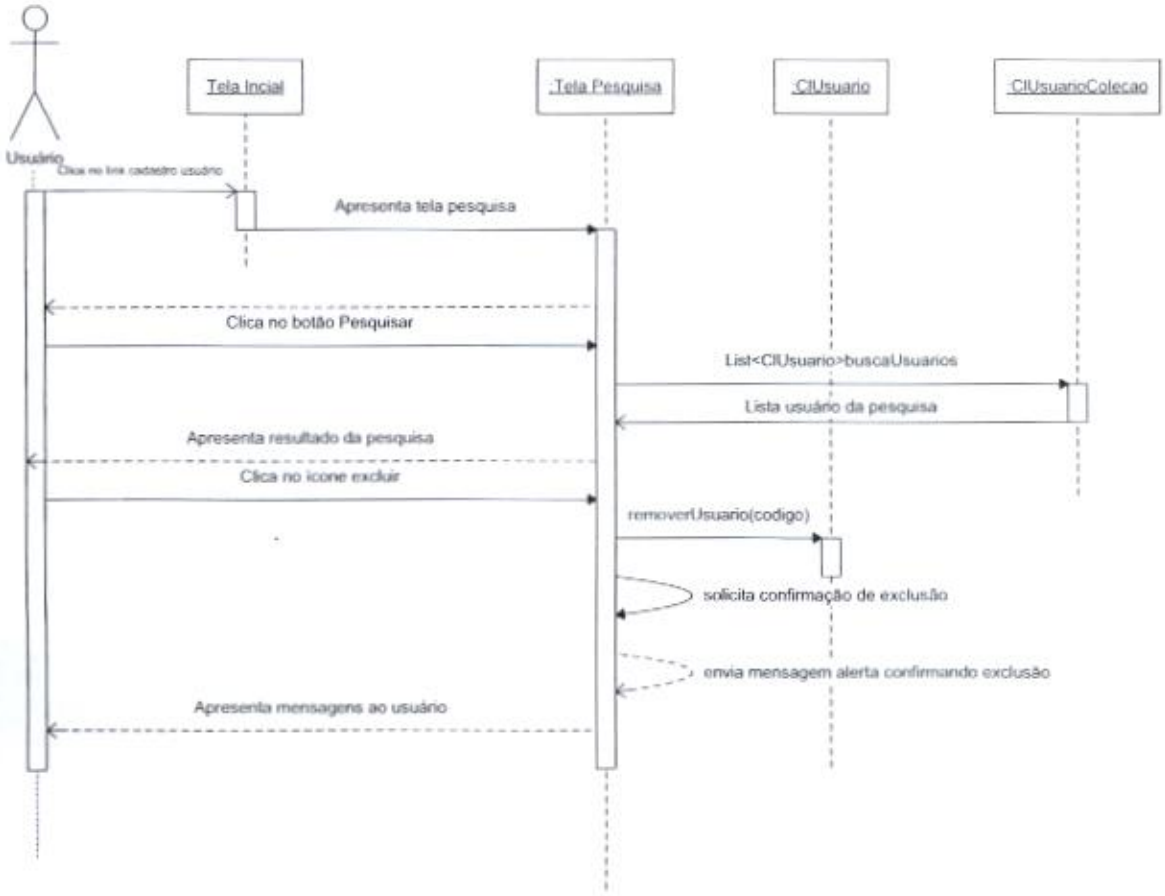
13.13 Diagrama de atividades – Alterar



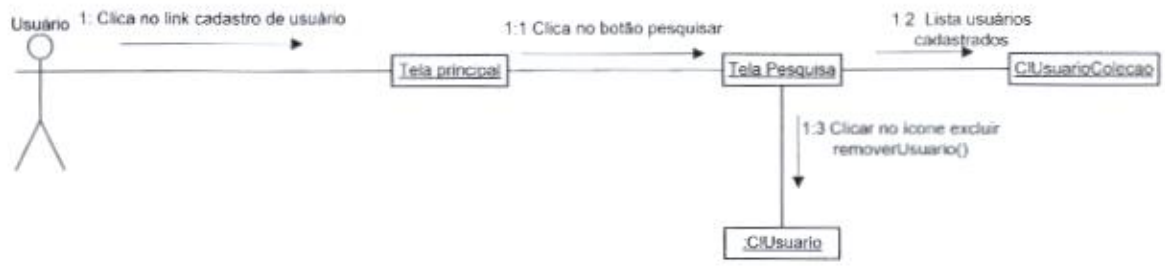
13.14 Diagrama de telas – Excluir



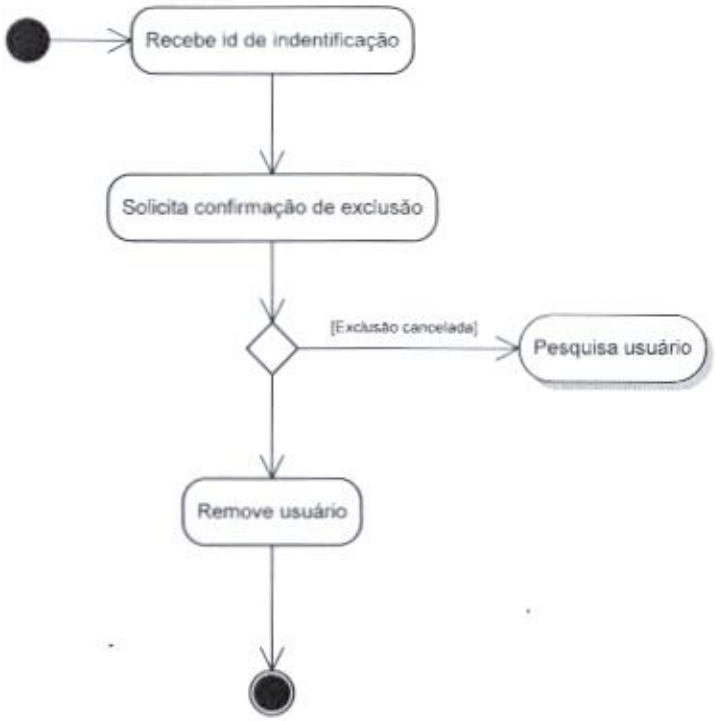
13.15 Diagrama de seqüência – Excluir



13.16 Diagrama de comunicação – Excluir



13.17 Diagrama de atividades – Excluir



14. CADASTRAR CONFIGURAÇÕES

14.1 Evento de caso de uso

Pré-requisitos: nenhum

Fluxo Principal

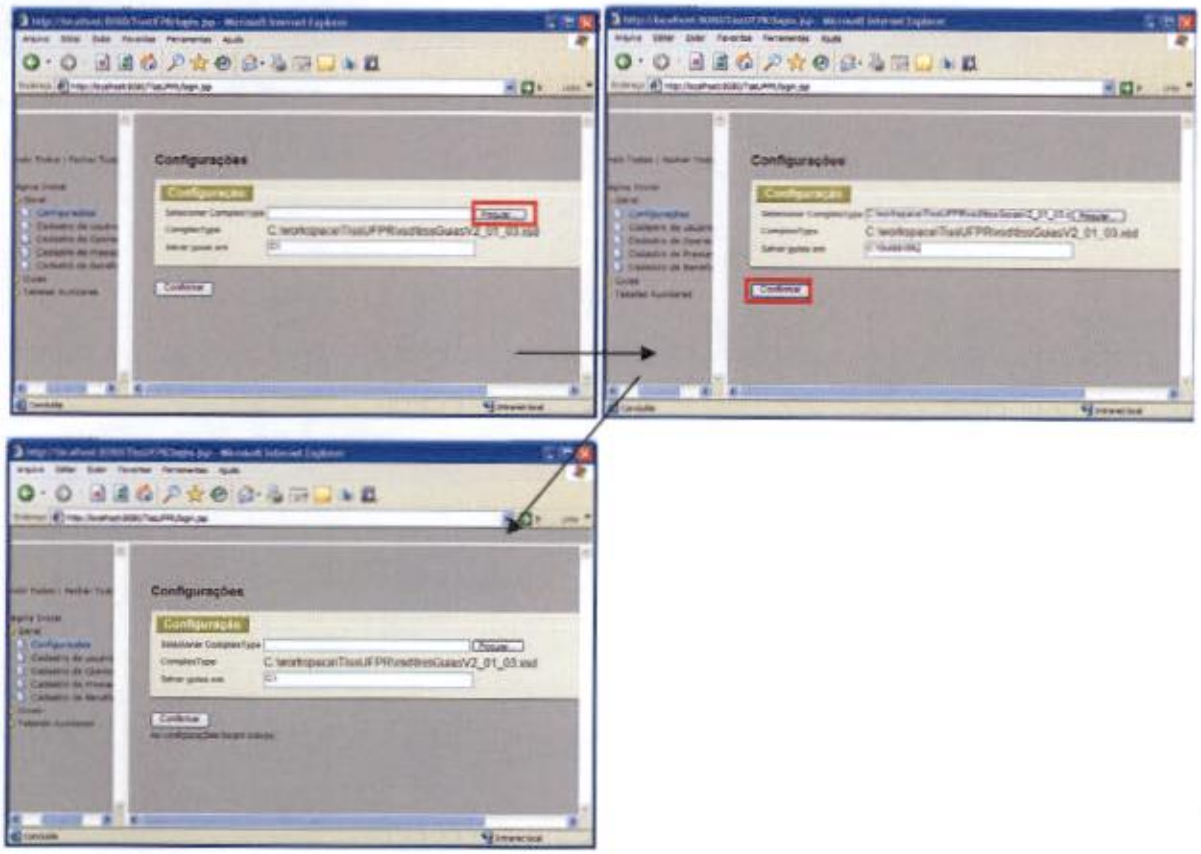
O administrador seleciona o link configurações. No campo indicado ele coloca o caminho onde se encontra o diretório do arquivo.xsd dos *complex types* das guias possibilitando assim que o sistema leia o arquivo.

Nessa mesma tela, o administrador indicará o local de armazenamento das guias no modelo XML gerado.

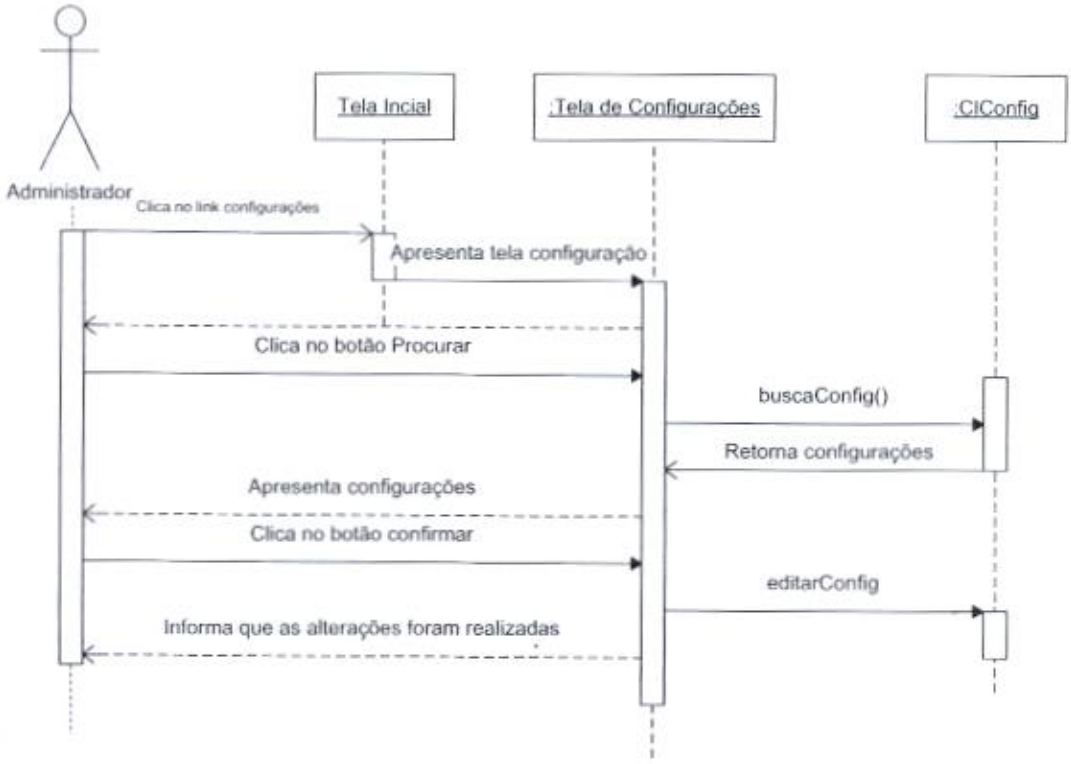
Fluxo alternativo

Para esse caso de uso, não possuímos fluxo alternativo.

14.2 Diagrama de telas

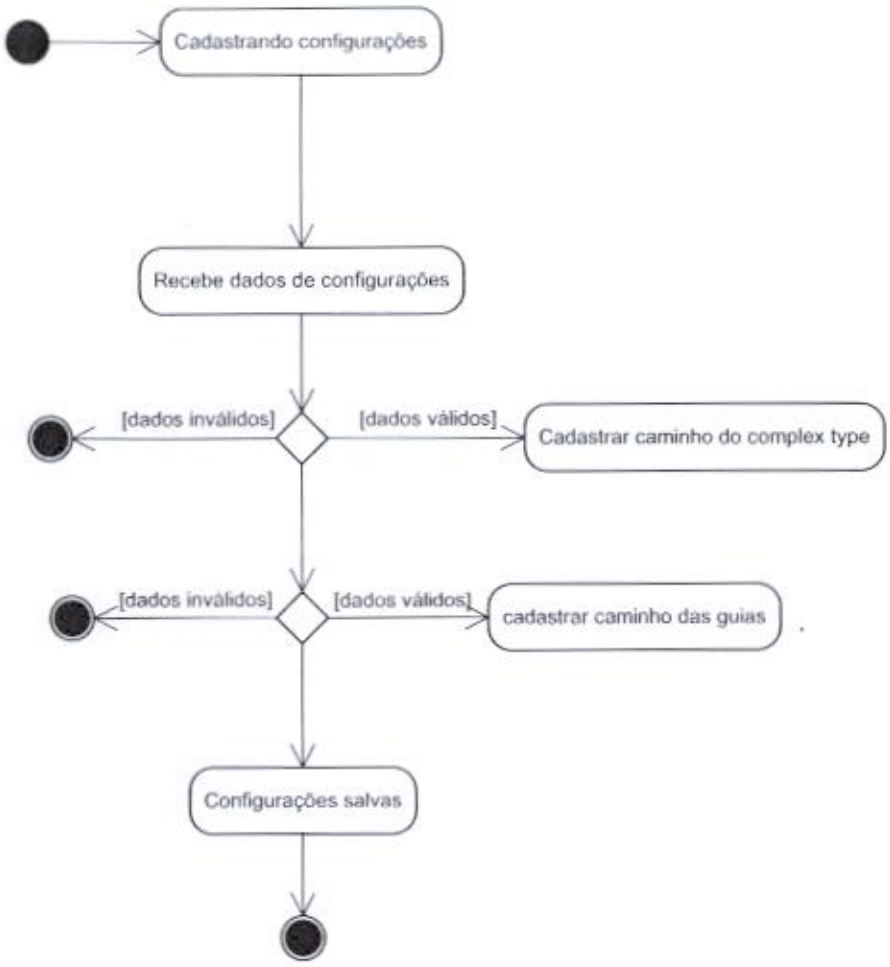


14.3 Diagrama de seqüência



14.4 Diagrama de comunicação

14.5 Diagrama de atividades



15. CADASTRAR OPERADORAS

15.1 Evento do caso de uso

Pré-requisitos: nenhum

Fluxo Principal

O usuário seleciona o link de cadastro de operadora. A tela inicial do cadastro de operadora é a de pesquisa. Nessa tela, o usuário clica no botão "Novo cadastro" que trará a tela de cadastro. Após digitar todos os dados do cadastro, o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro que acabou de ser realizado, indicando assim que foi incluso o novo cadastro no banco.

Fluxos alternativos

Pesquisar

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e clica no botão "Pesquisar". O sistema listará todos os cadastros de operadoras, que atendam os parâmetros do filtro.

Alterar

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e clica no botão "Pesquisar". O sistema listará todos os cadastros nos quais os nomes das operadoras atendam os parâmetros desejados. Na lista gerada, o usuário clica no ícone alterar ( ) referente ao cadastro que deseja alterar, abrirá a tela para a alteração dos dados. Tendo lançado todos os dados que se deseja alterar o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro do beneficiário que acabou de ser alterado, indicando assim que foram realizas as devidas alterações no cadastro.

Excluir

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e clica no botão "Pesquisar". O sistema listará todos os nomes com os parâmetros desejados. Na lista gerada, o usuário clica no icone excluir ( ) referente ao cadastro que deseja excluir. Aparecerá um alerta solicitando a confirmação da exclusão, o usuário clicando no botão "OK", confirmando a exclusão.

Cancelar

Se o usuário clicar em "Cancelar", o sistema interromperá o cadastro.

Cadastrar campos obrigatórios na operadora

Pré-requisito: Esse caso de uso está incluso no caso de uso cadastrar operadora.

Fluxo Principal

No momento de cadastro da operadora, o usuário seleciona nas abas correspondentes a cada guia, os campos opcionais que, para essa operadora são de envio obrigatório quando na geração do XML.

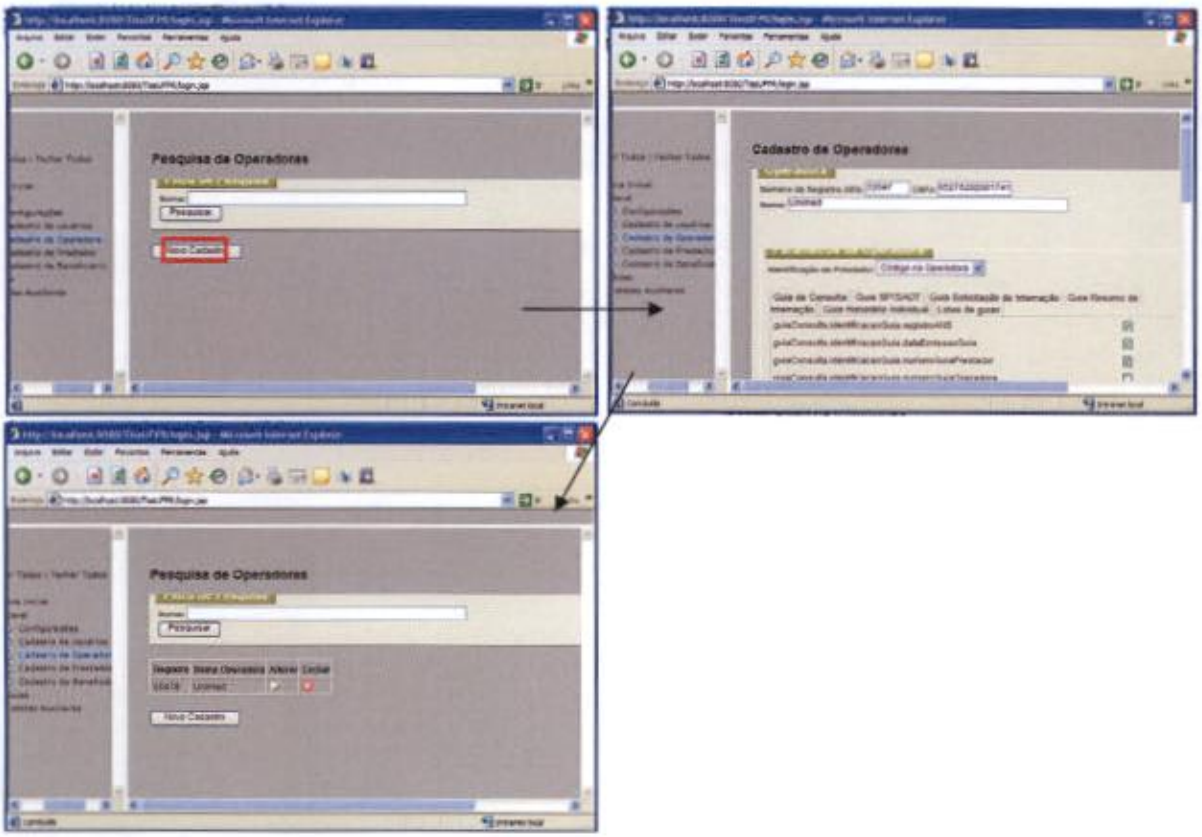
Cadastrar lotes

Pré-requisito: Ser cadastrado antes de iniciar a geração de guias para essa operadora. Esse caso de uso é uma inclusão do caso cadastrar operadora.

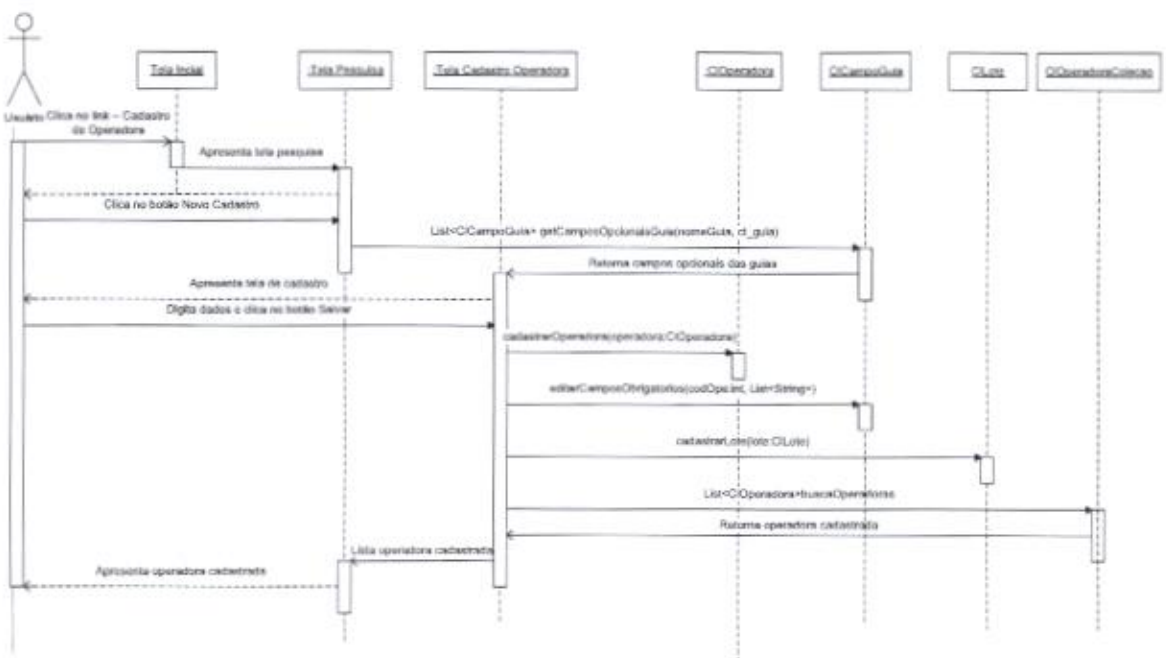
Fluxo Principal

No momento de cadastrar a operadora, o usuário seleciona a aba lotes e cadastra o número inicial do lote e o número final do lote das guias. O sistema irá disponibilizar os números de cada guia seqüencialmente.

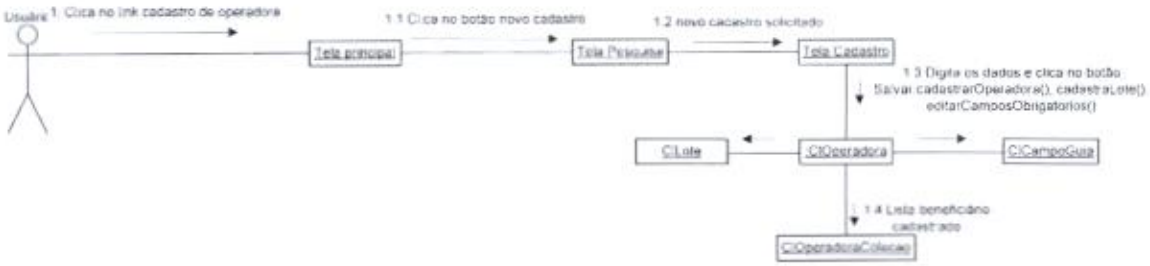
15.2 Diagrama de telas - Cadastrar



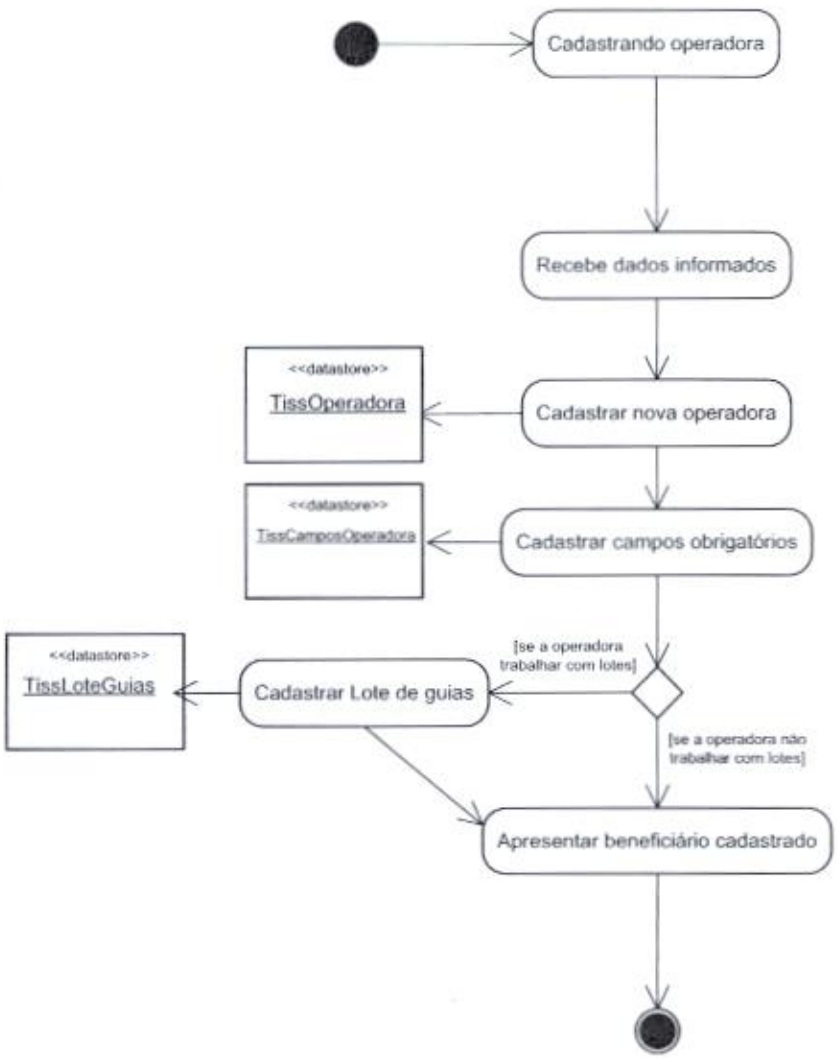
15.3 Diagrama de seqüência – Cadastrar



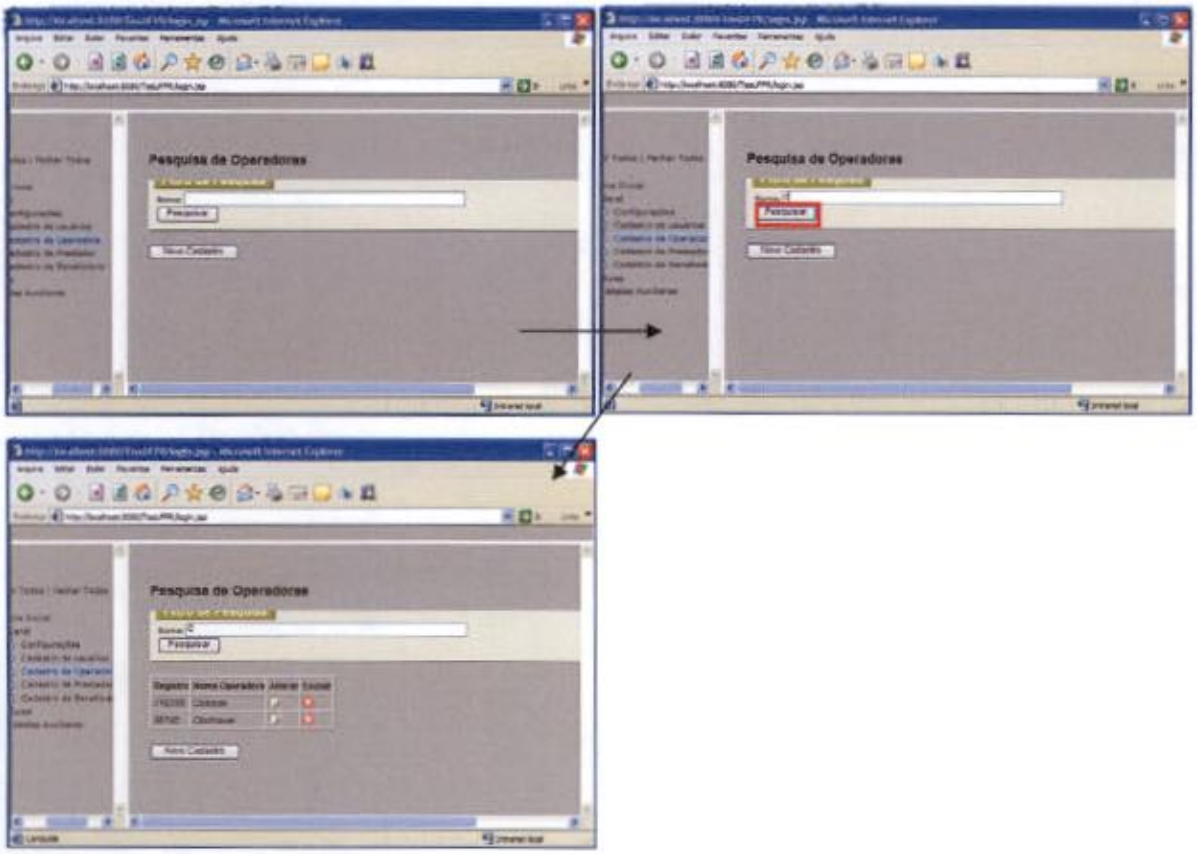
15.4 Diagrama de comunicação – Cadastrar



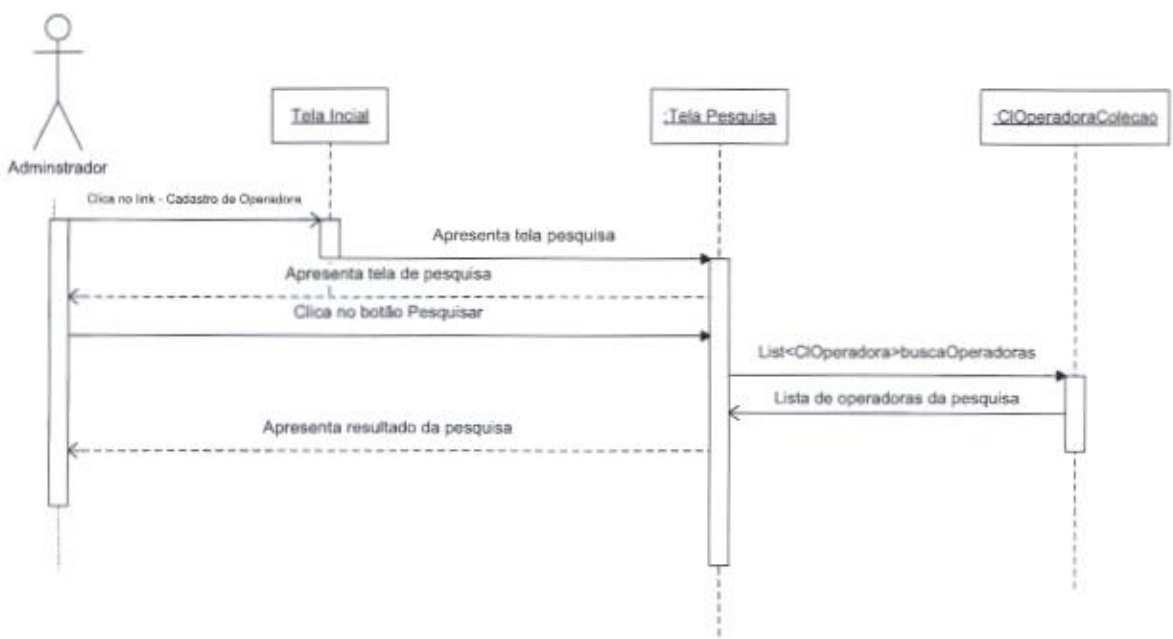
15.5 Diagrama de atividades – Cadastrar



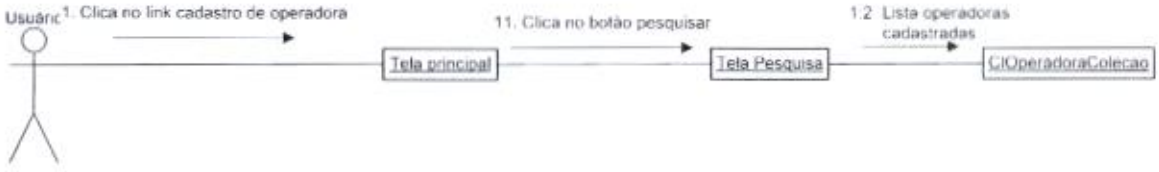
15.6 Diagrama de telas – Pesquisar



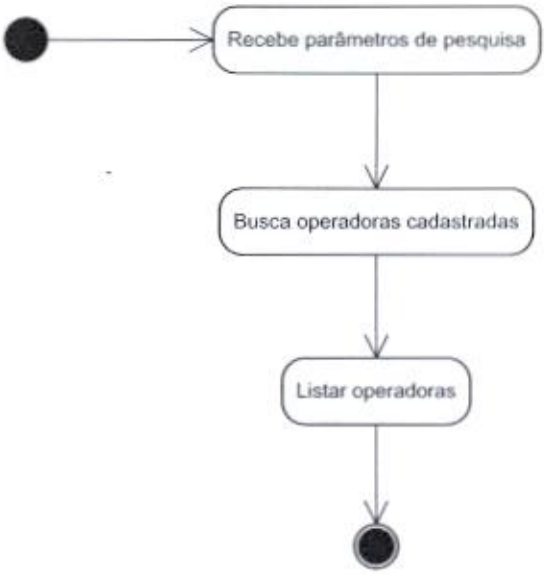
15.7 Diagrama de seqüência – Pesquisar



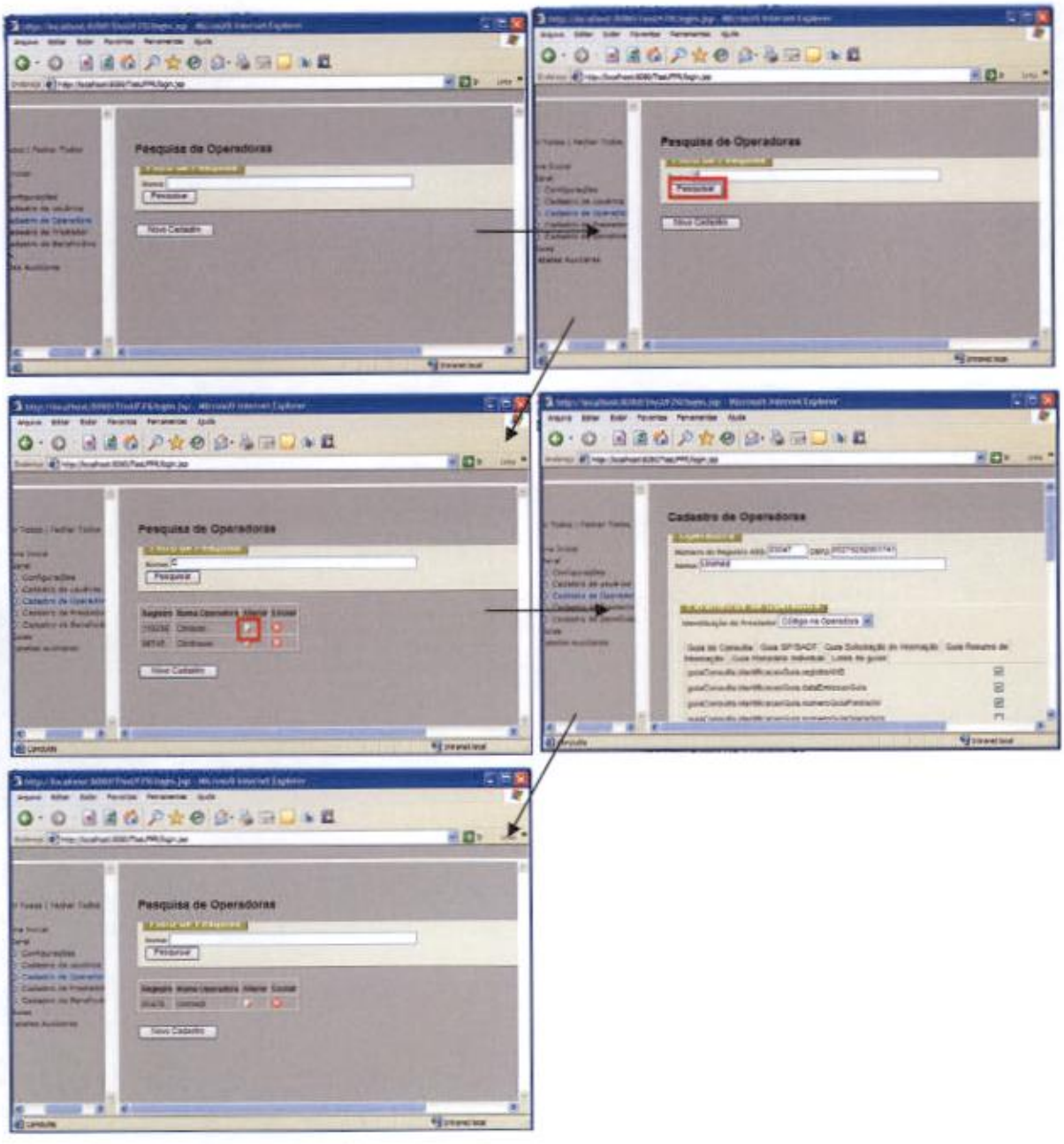
15.8 Diagrama de comunicação – Pesquisar



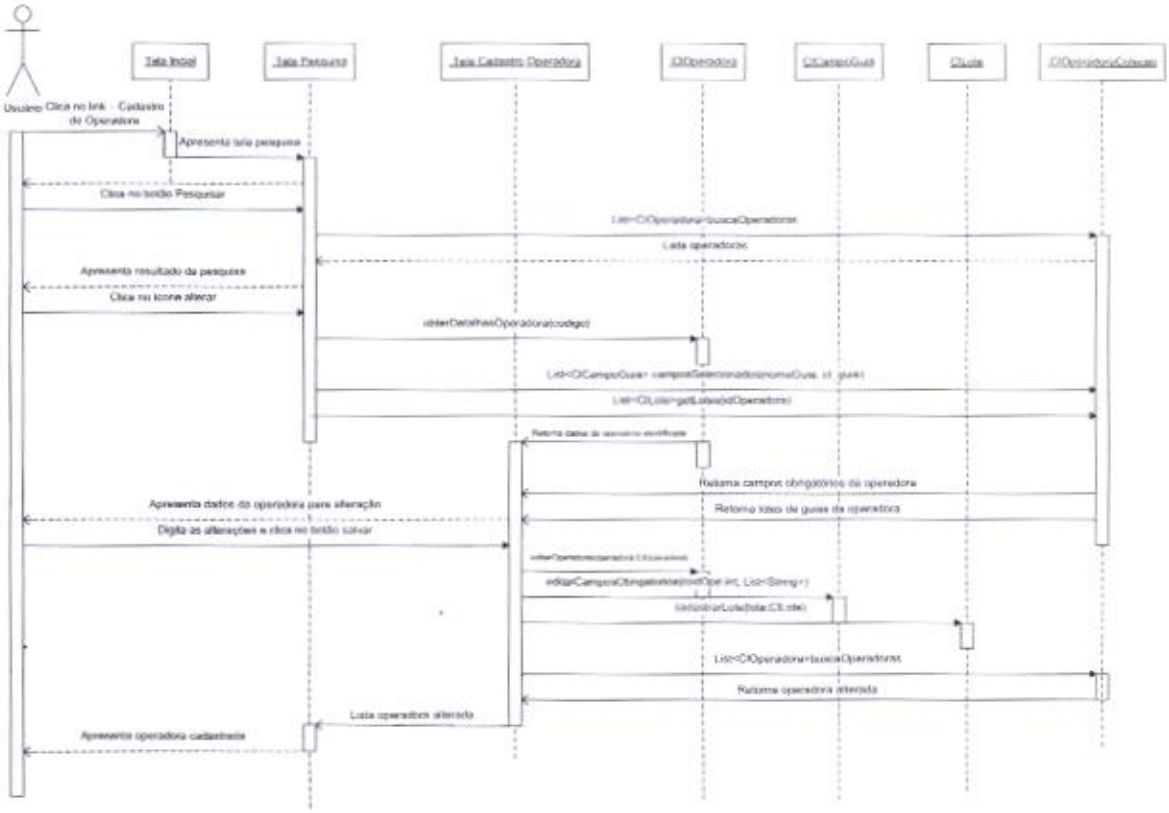
15.9 Diagrama de atividades – Pesquisar



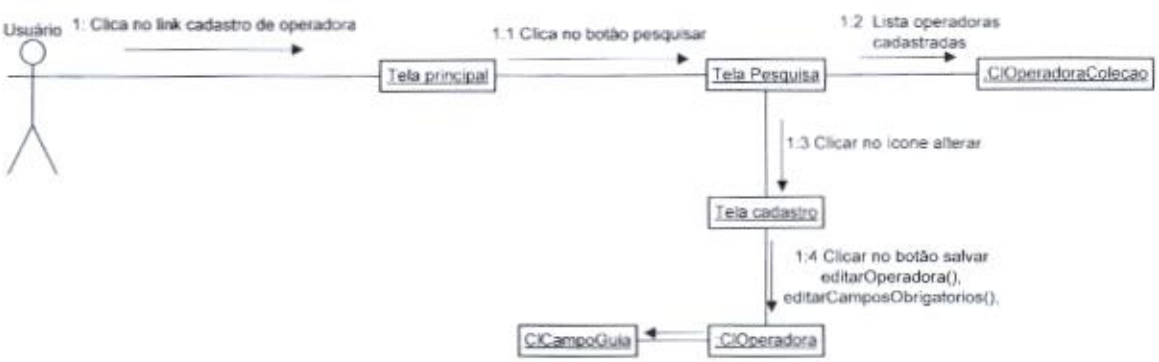
15.10 Diagrama de telas – Alterar



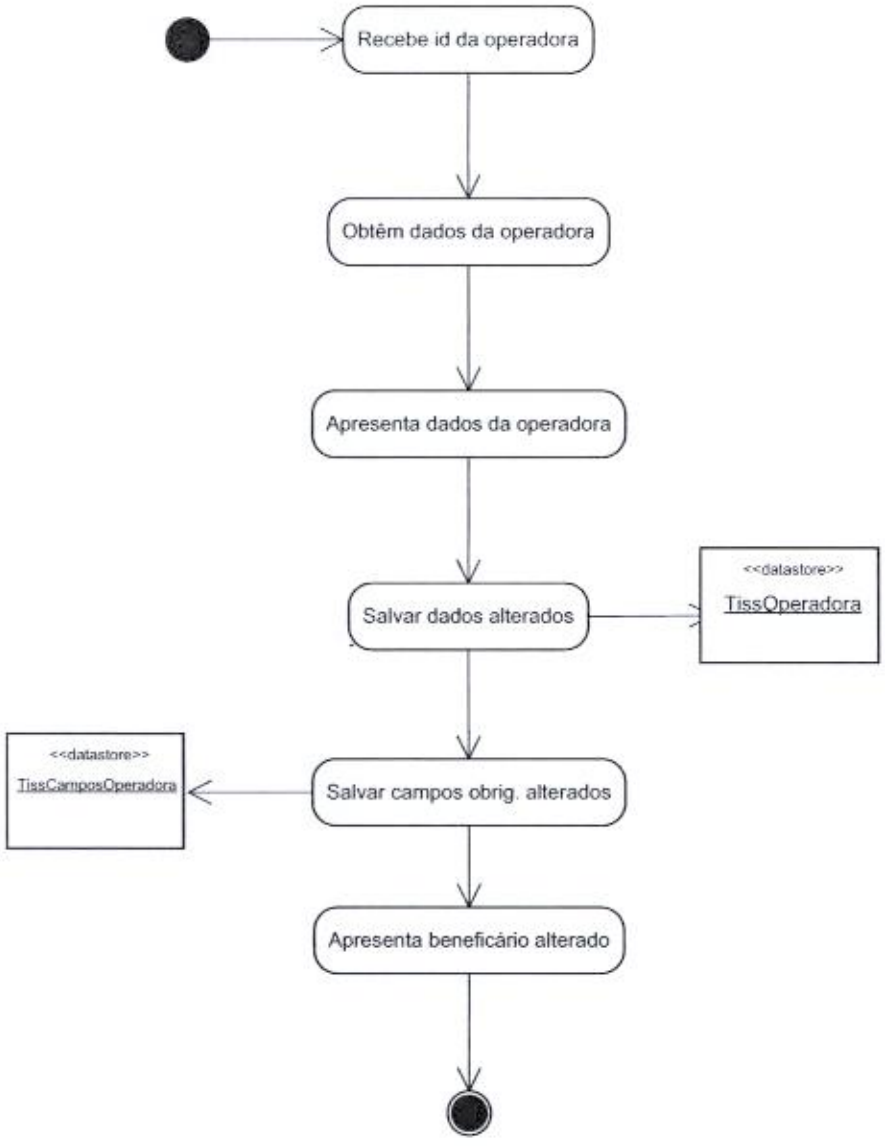
15.11 Diagrama de seqüência – Alterar



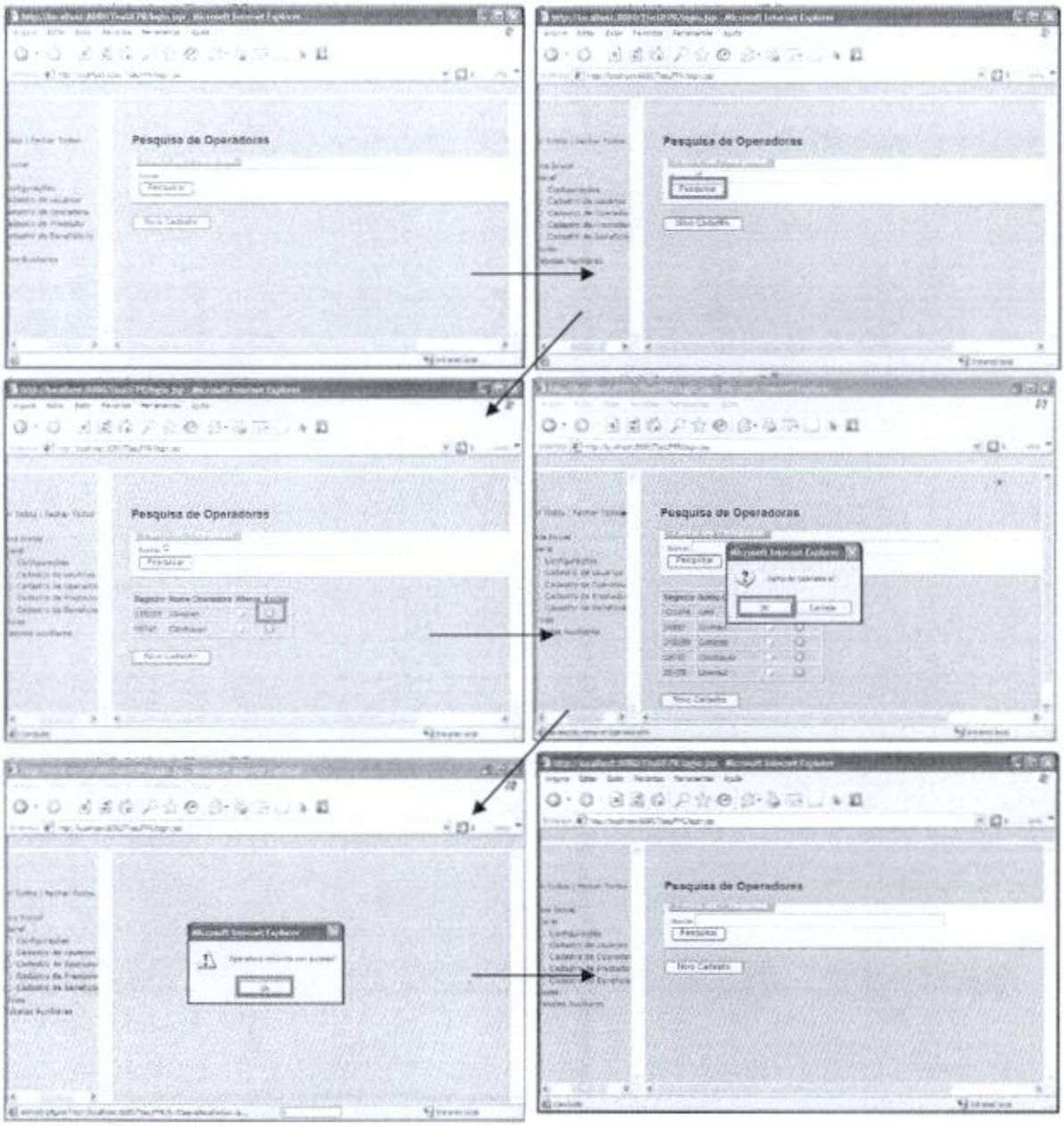
15.12 Diagrama de comunicação – Alterar



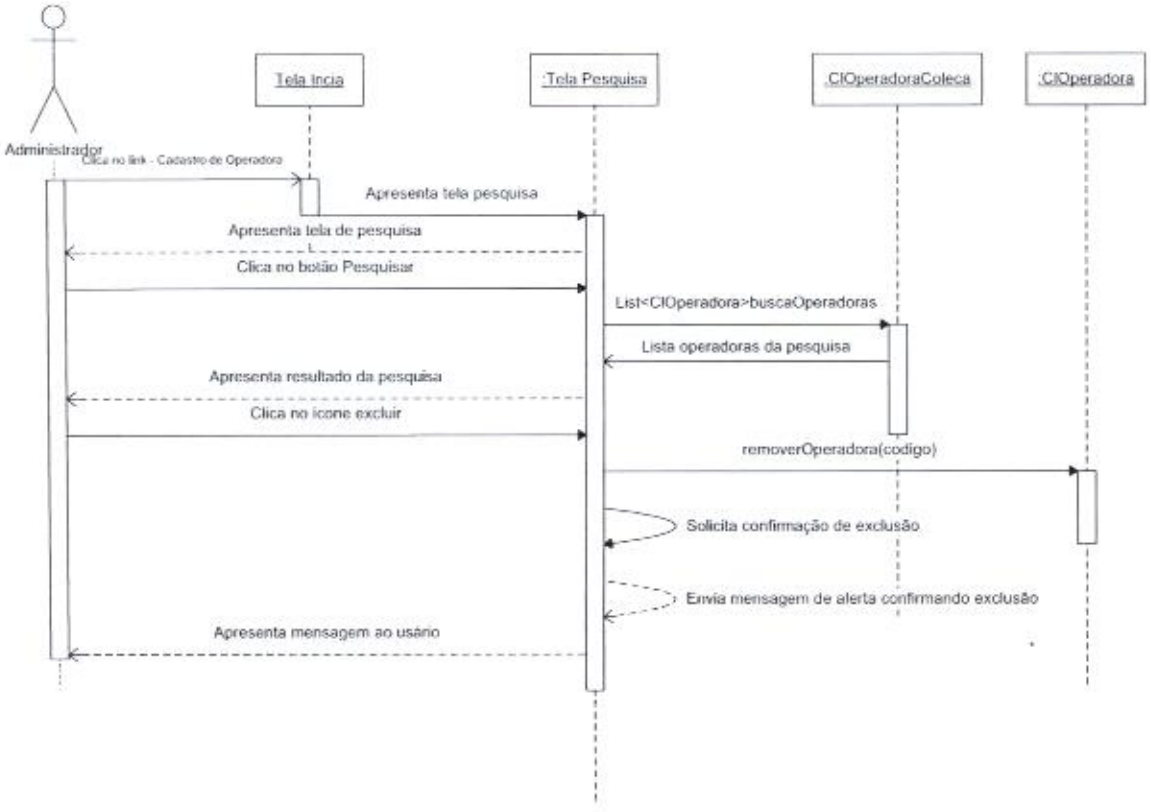
15.13 Diagrama de atividades – Alterar



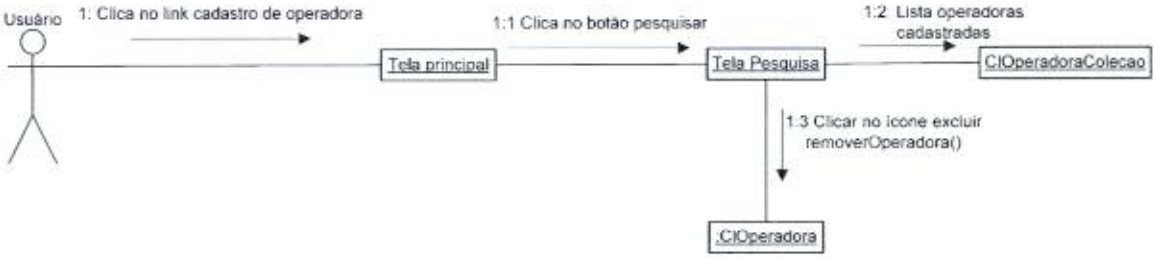
15.14 Diagrama de telas – Excluir



15.15 Diagrama de seqüência – Excluir



15.16 Diagrama de comunicação – Excluir



15.17 Diagrama de atividades – Excluir

16. CADASTRAR BENEFICIÁRIOS

16.1 Evento do caso de uso

Pré-requisitos: nenhum

Fluxo Principal

O usuário seleciona o link de cadastro de beneficiários. A tela inicial do cadastro de beneficiários é a de pesquisa. Nessa tela, o usuário clica no botão "Novo cadastro" que trará a tela de cadastro. Após digitar todos os dados do cadastro, o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro que acabou de ser realizado, indicando assim que foi incluso o novo cadastro.

Fluxos alternativos

Pesquisar

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e clica no botão "Pesquisar". O sistema listará todos os cadastros de beneficiários, que atendam os parâmetros do filtro.

Alterar

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e clica no botão "Pesquisar". O sistema listará todos os nomes com os parâmetros desejados. Na lista gerada, o usuário clica no icone alterar ( ) referente ao cadastro que deseja alterar, abrirá a tela para a alteração dos dados. Tendo lançado todos os dados que se deseja alterar o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro do beneficiário que acabou de ser alterado, indicando assim que foram realizas as devidas alterações no cadastro.

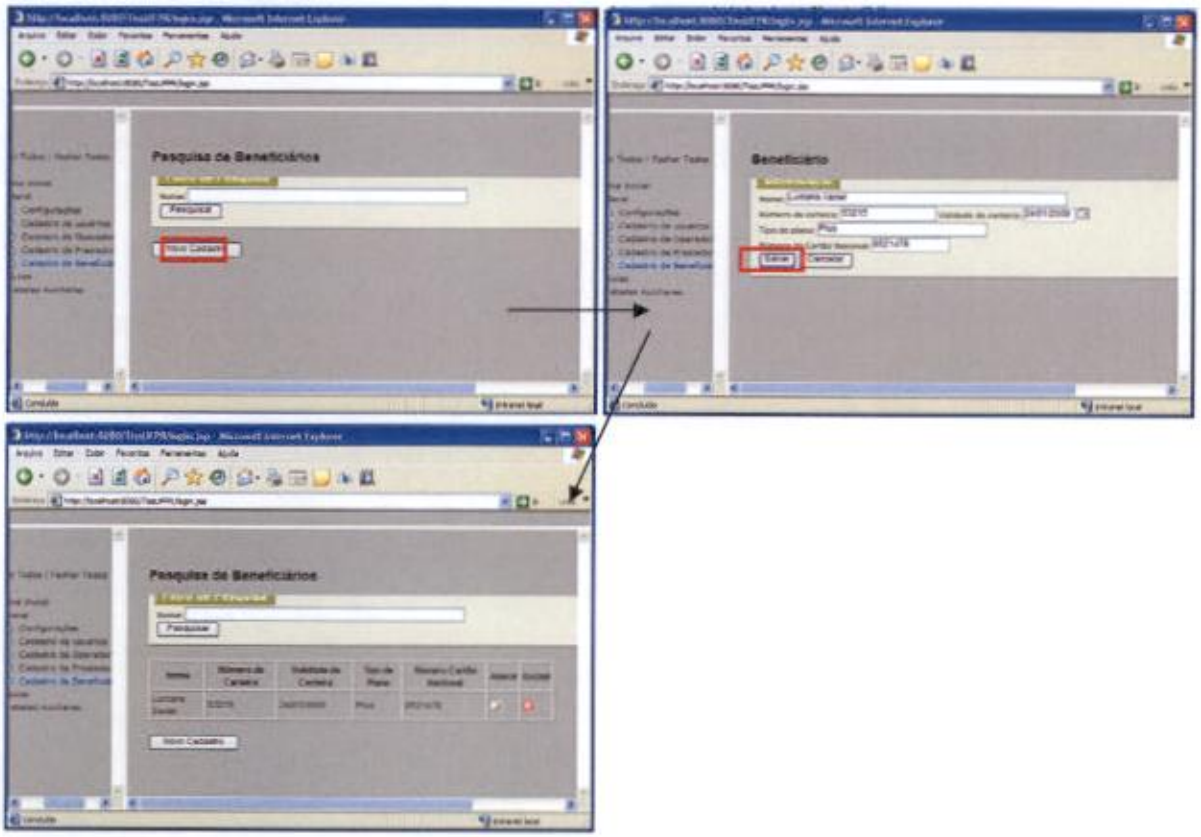
Excluir

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e clica no botão "Pesquisar". O sistema listará todos os nomes com os parâmetros desejados. Na lista gerada, o usuário clica no ícone excluir ( ) referente ao cadastro que deseja excluir. Aparecerá um alerta solicitando a confirmação da exclusão, o usuário clicando no botão "OK", confirmando a exclusão.

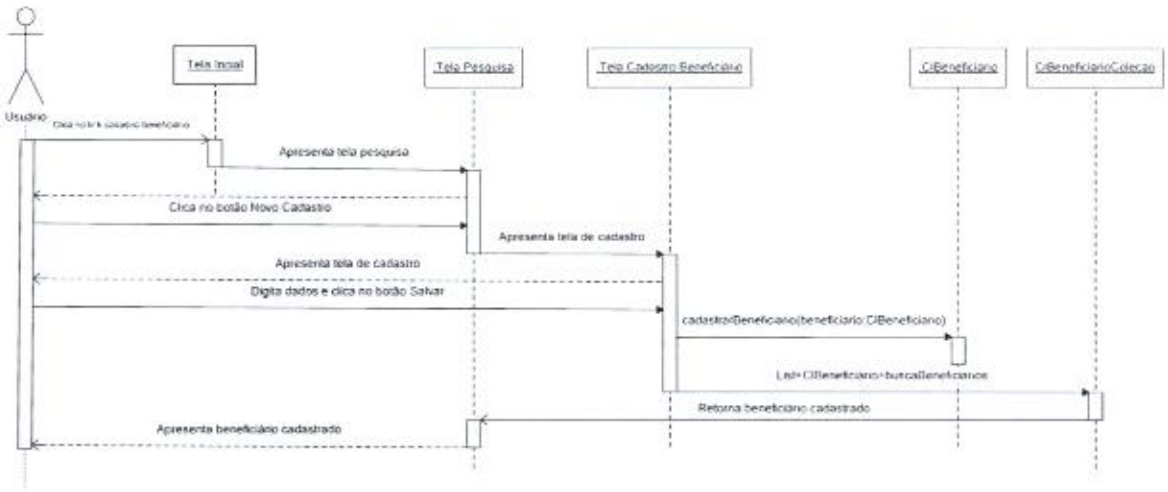
Cancelar

Se o usuário clicar em "Cancelar", o sistema interromperá o cadastro.

16.2 Diagrama de telas – Cadastrar

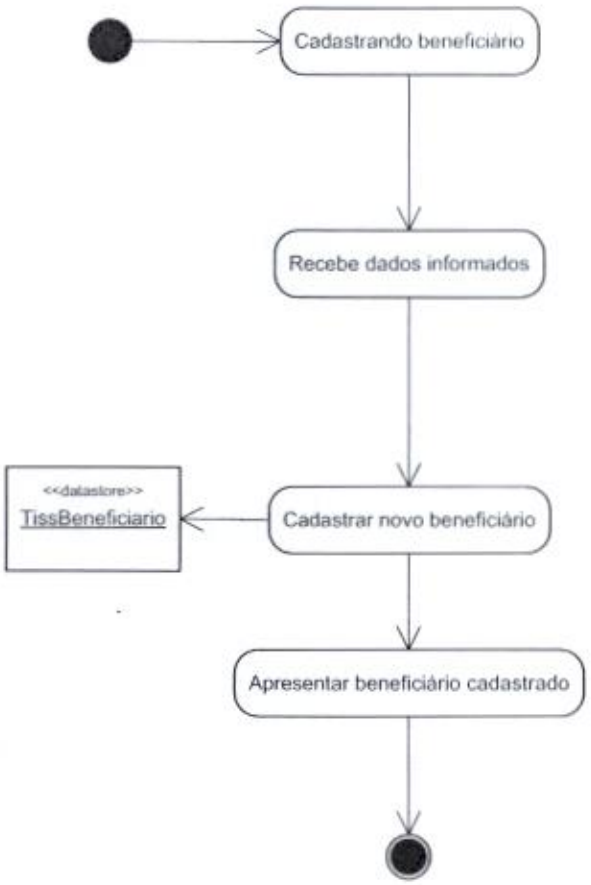


16.3 Diagrama de seqüência – Cadastrar

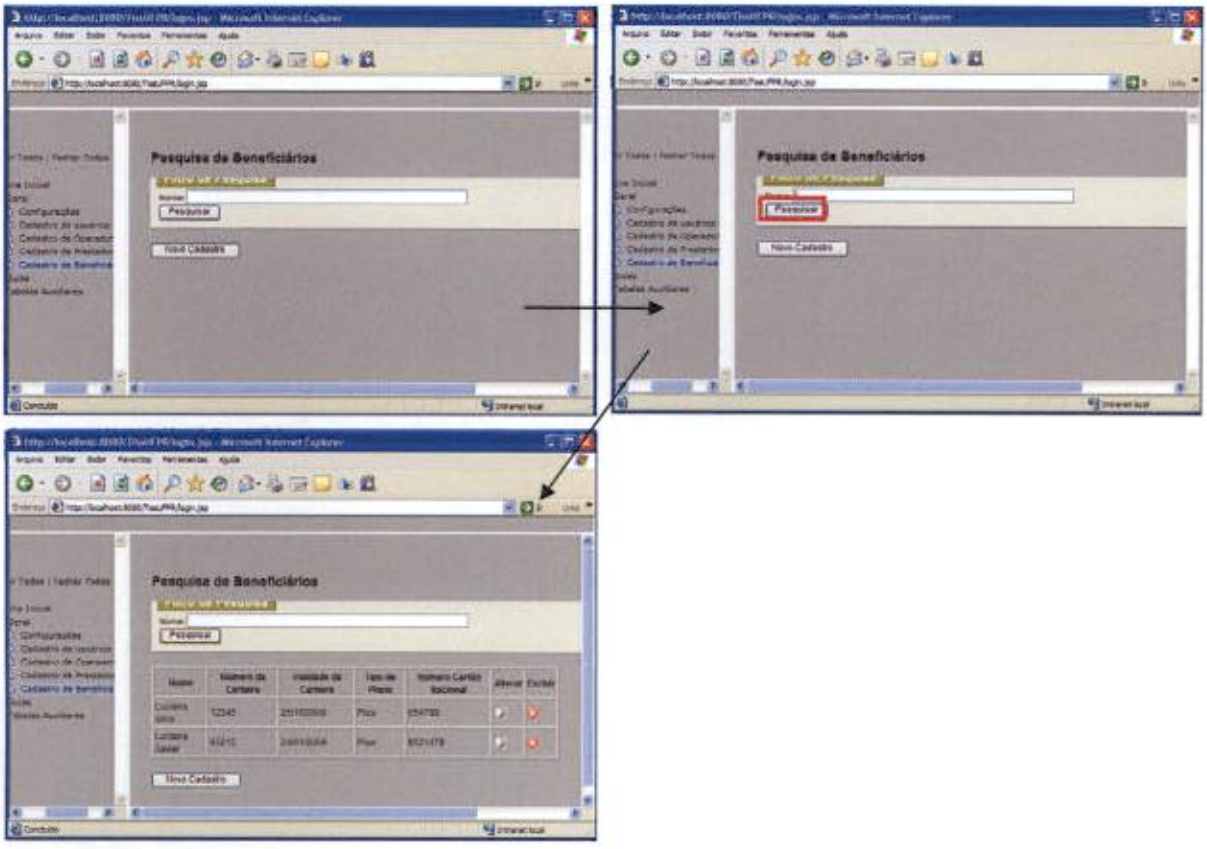


16.4 Diagrama de comunicação – Cadastrar

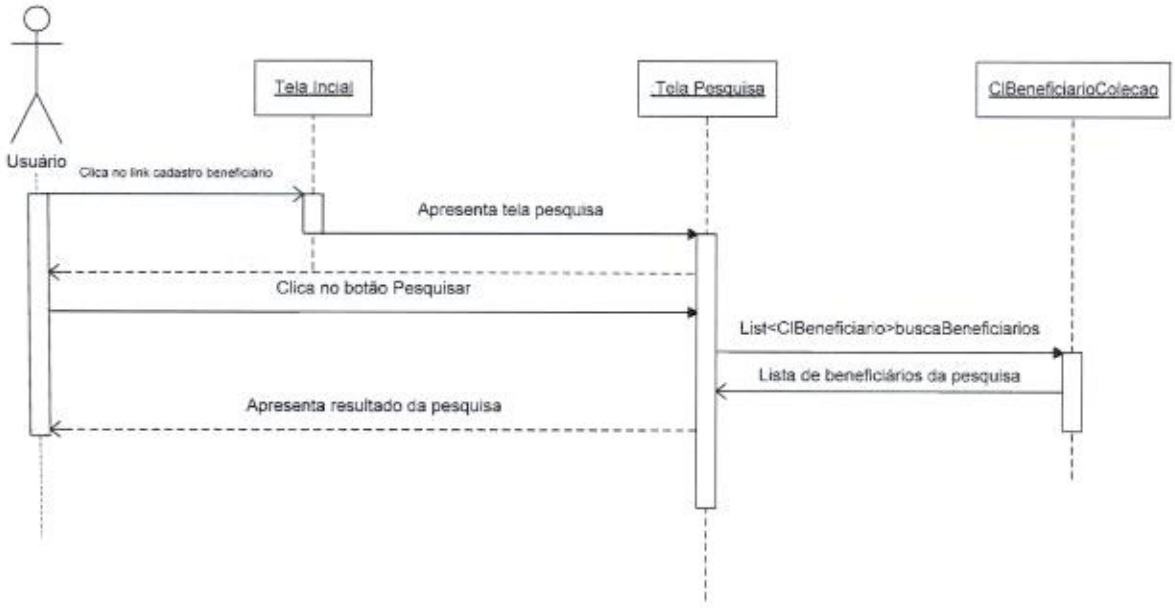
16.5 Diagrama de atividades – Cadastrar



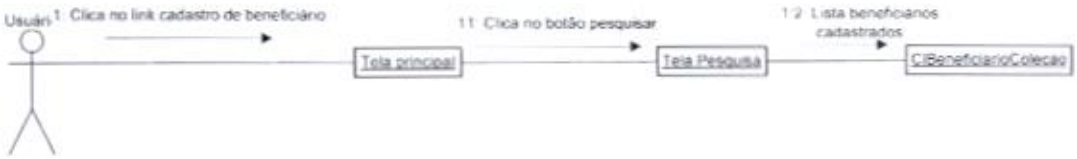
16.6 Diagrama de telas – Pesquisar



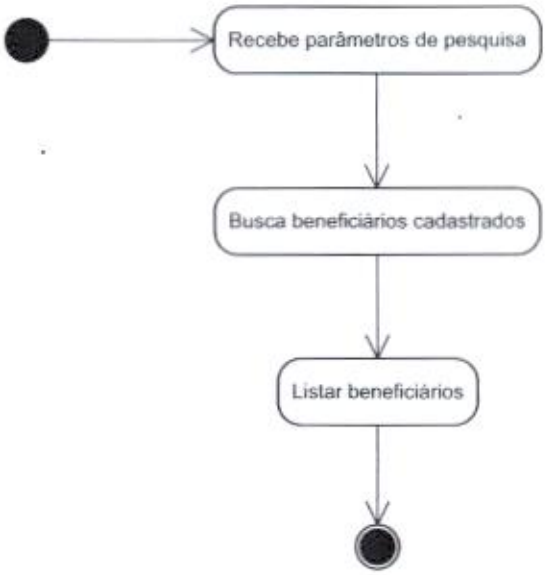
16.7 Diagrama de seqüencia – Pesquisar



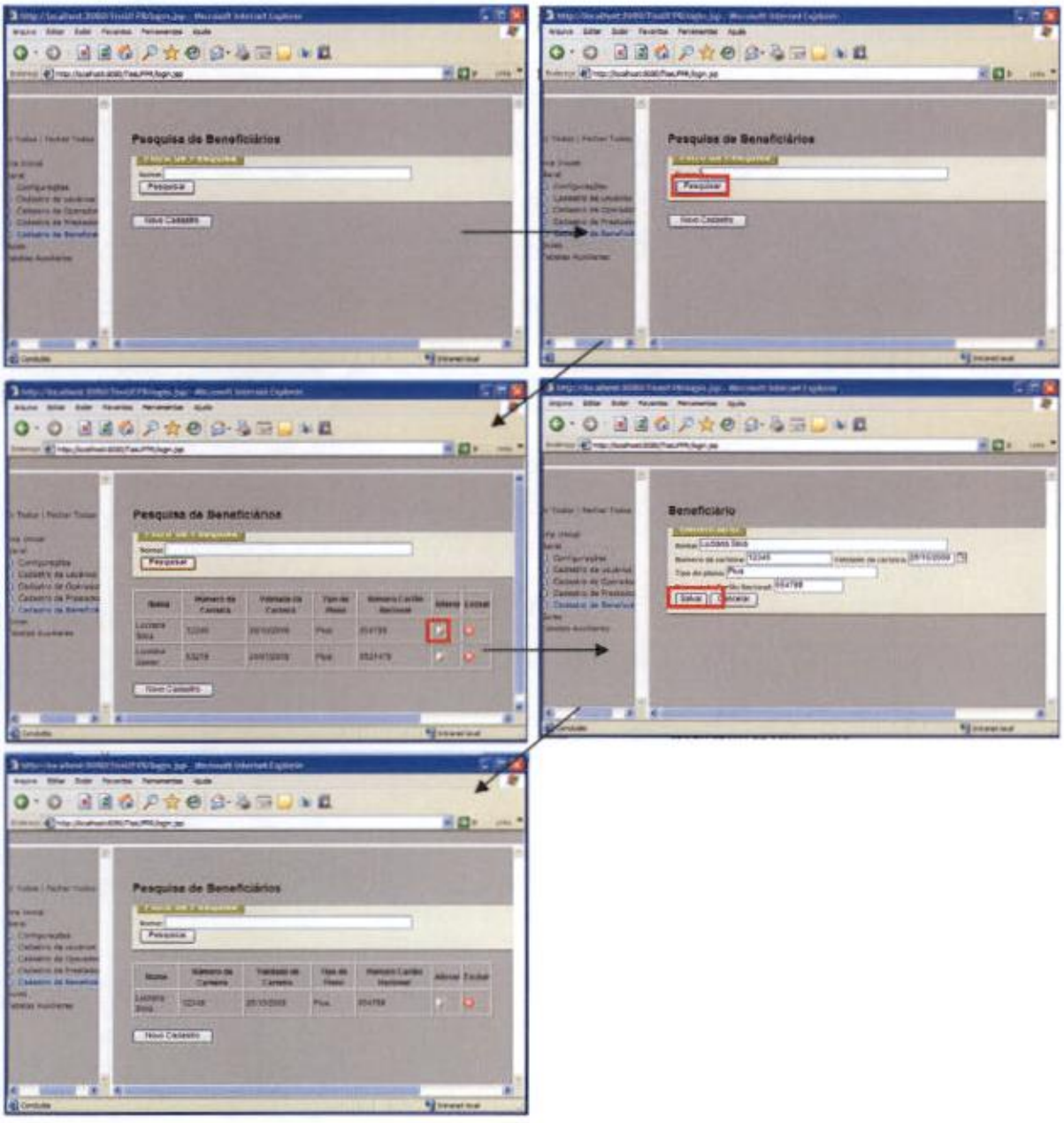
16.8 Diagrama de comunicação – Pesquisar



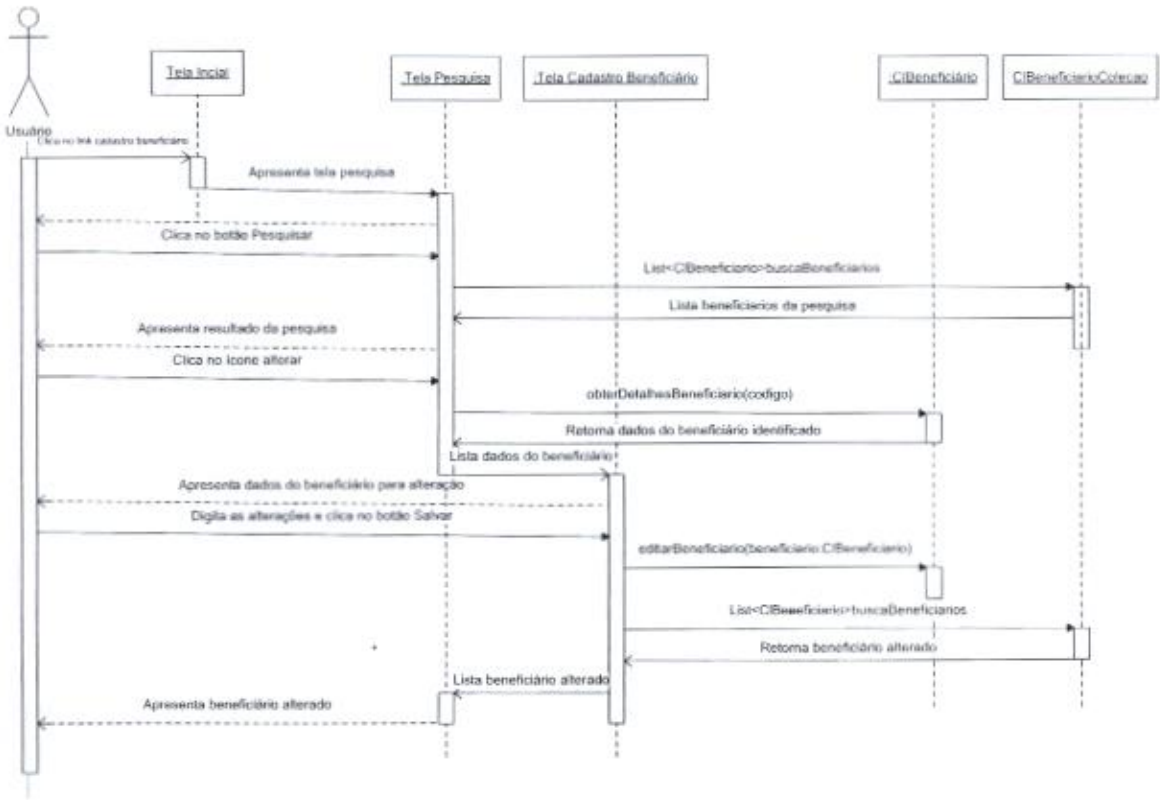
16.9 Diagrama de atividades – Pesquisar



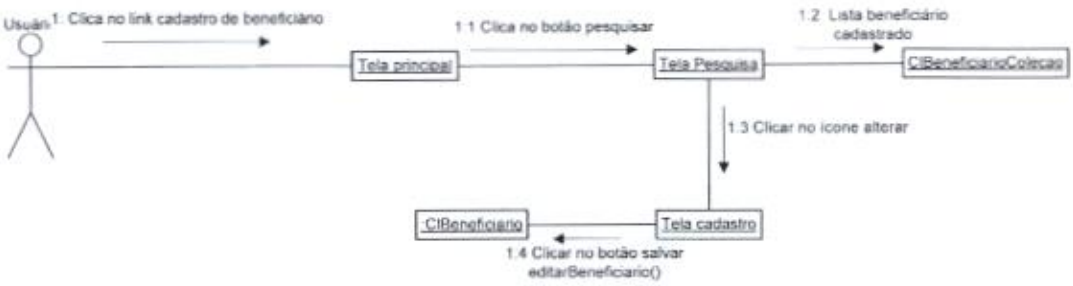
16.10 Diagrama de telas – Alterar



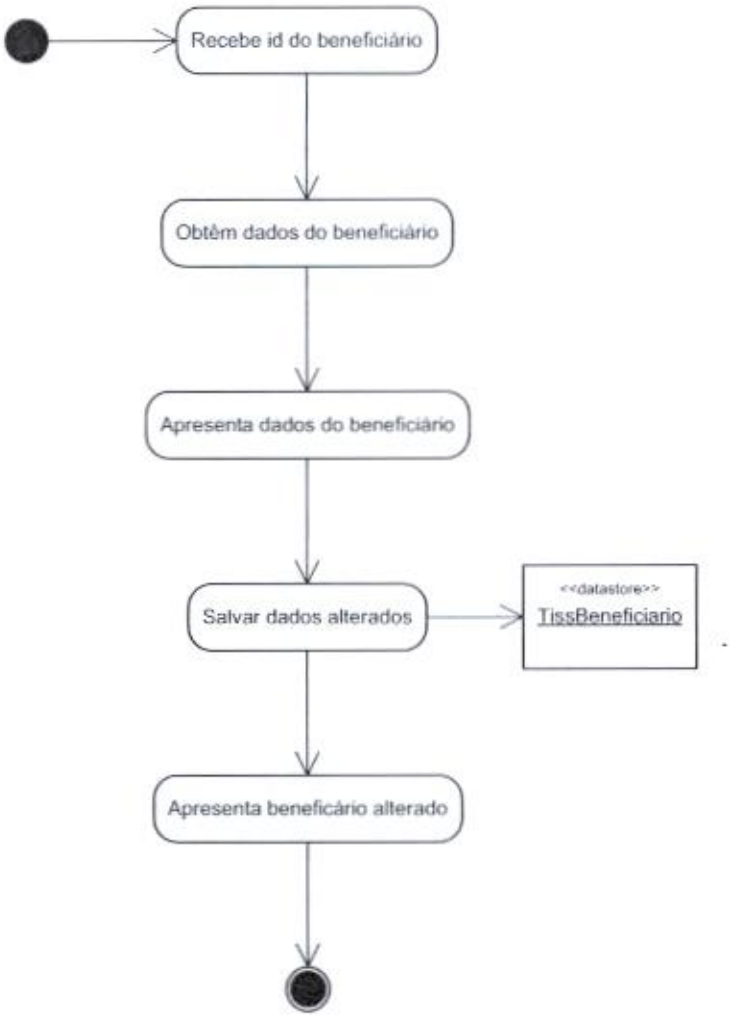
16.11 Diagrama de seqüência – Alterar



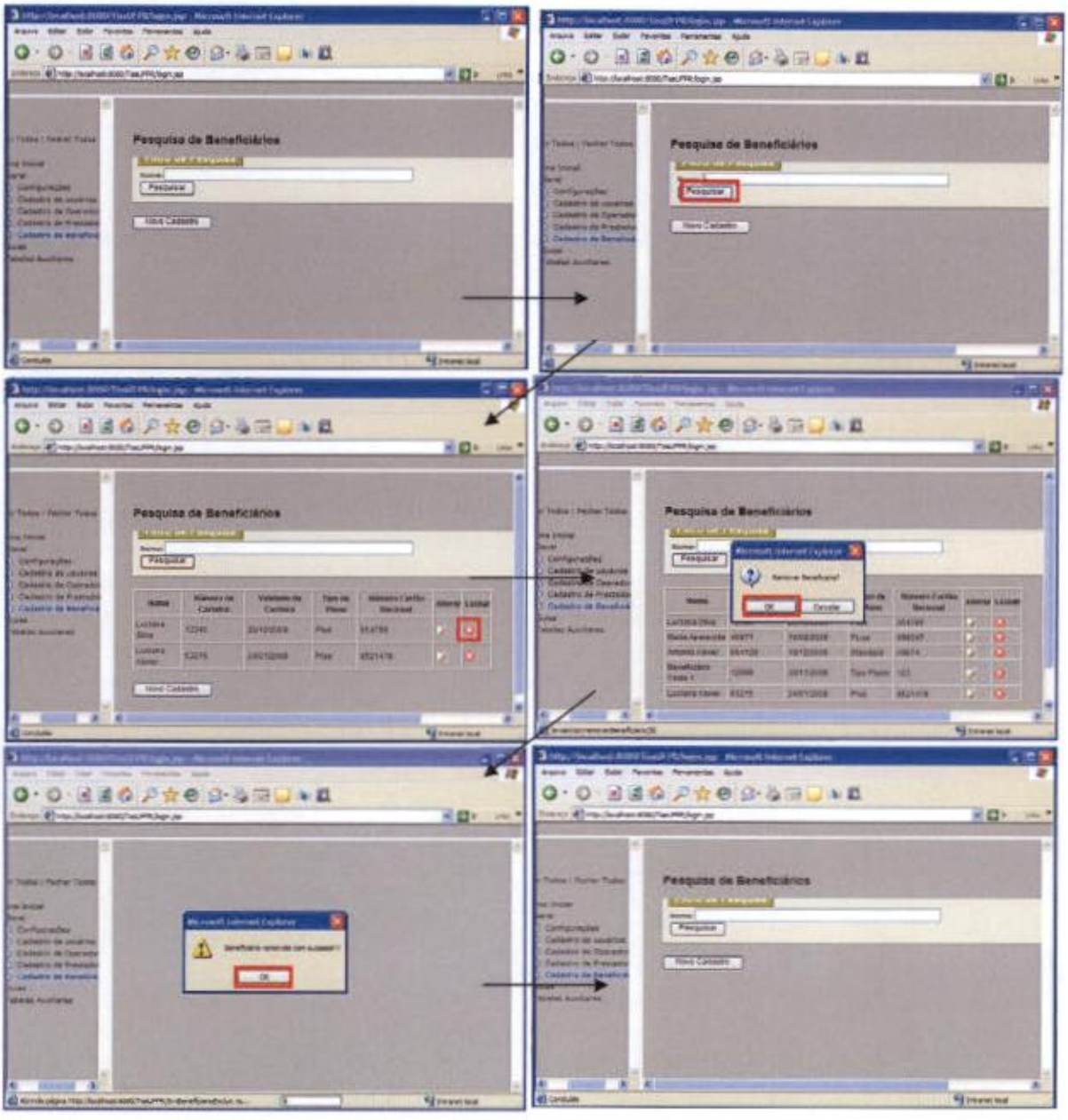
16.12 Diagrama de comunicação – Alterar



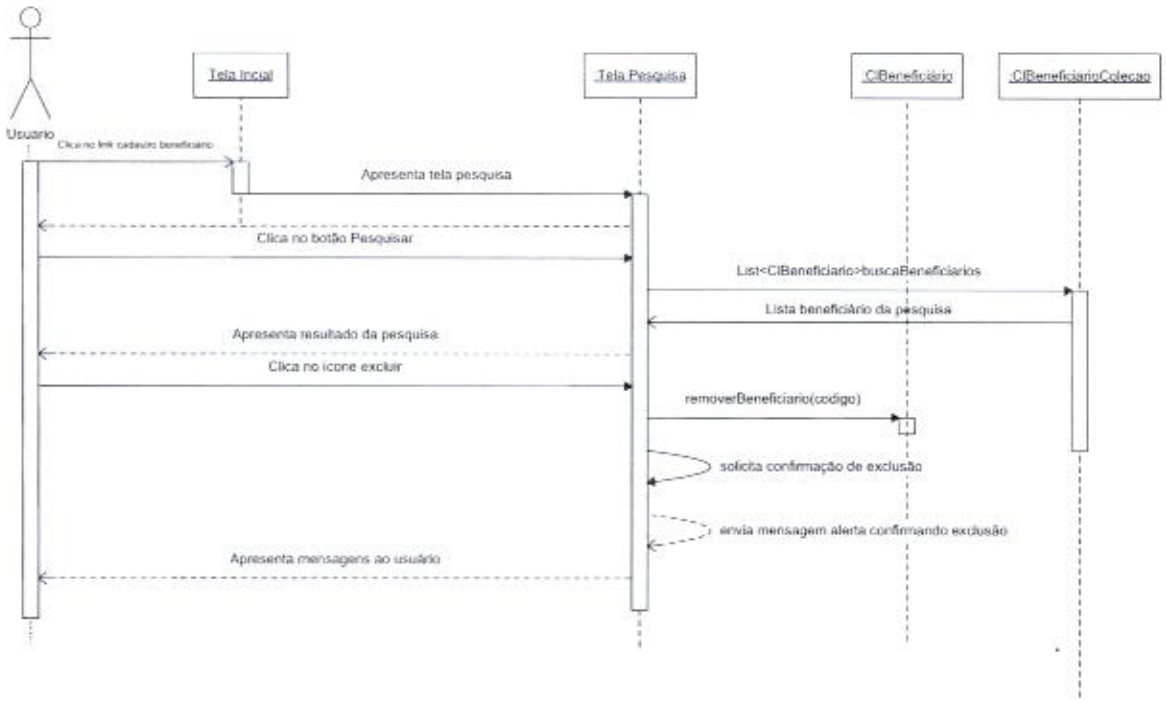
16.13 Diagrama de atividades – Alterar



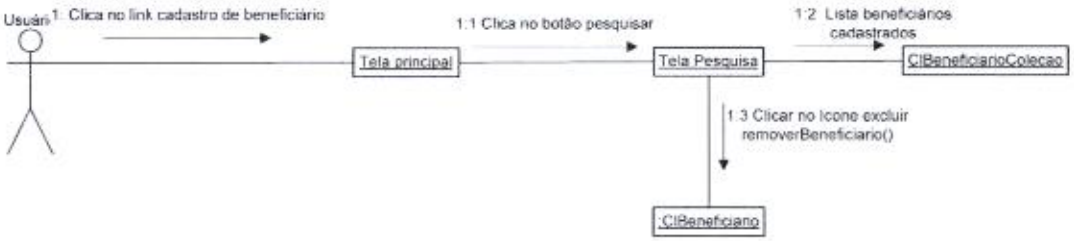
16.14 Diagrama de telas – Excluir



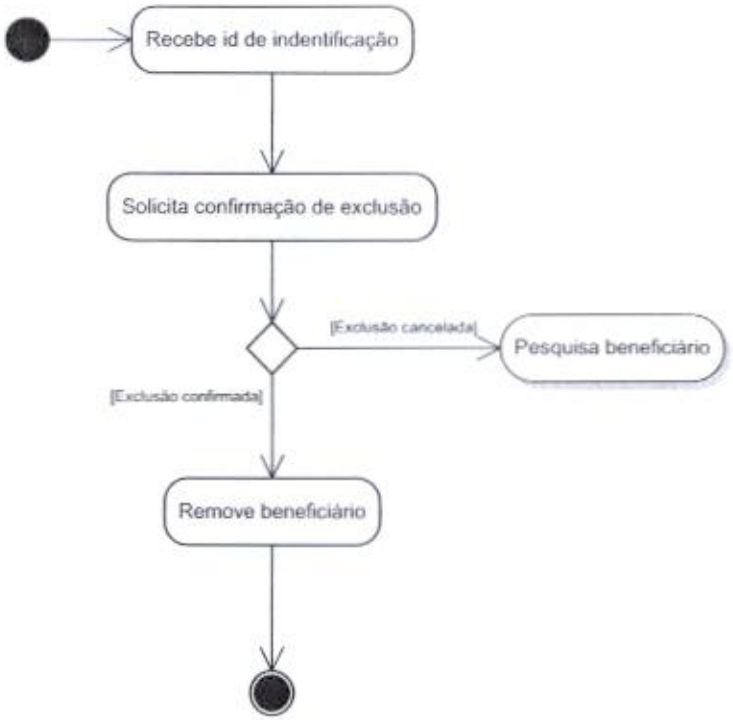
16.15 Diagrama de seqüência – Excluir



16.16 Diagrama de comunicação – Excluir



16.17 Diagrama de atividades - Excluir



17. CADASTRAR PRESTADORES

17.1 Eventos do caso de uso

Pré-requisitos: nenhum

Fluxo Principal

O usuário seleciona o link de cadastro de prestador. A tela inicial do cadastro de prestador é a de pesquisa. Clicando no botão "Novo cadastro", abrirá um pop-up na própria tela de pesquisa, onde o usuário deverá digitar o CNPJ ou CPF para validação inicial, após a validação o sistema trará a tela de cadastro. O usuário digita todos os dados do cadastro e clica no botão "Salvar". O sistema trará novamente a tela de pesquisa, listando somente o cadastro que acabou de ser realizado, indicando assim que foi incluso o novo cadastro no banco.

Fluxos alternativos

Pesquisar

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e/ou seleciona a opção de filtro para físico ou jurídico, e clica no botão "Pesquisar". O sistema listará todos os cadastros de prestadores, que atendam os parâmetros do filtro.

Alterar

Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e/ou seleciona a opção de filtro para físico ou jurídico, e clica no botão "Pesquisar". O sistema listará todos os cadastros nos quais os nomes dos prestadores atendam os parâmetros desejados. Na lista gerada, o usuário clica no ícone alterar ( ) referente ao cadastro que deseja alterar, abrirá a tela para a alteração dos dados. Tendo lançado todos os dados que

se deseja alterar o usuário clica no botão "Salvar" e o sistema trará novamente a tela de pesquisa, listando somente o cadastro do prestador que acabou de ser alterado, indicando assim que foram realizas as devidas alterações no cadastro.

Excluir

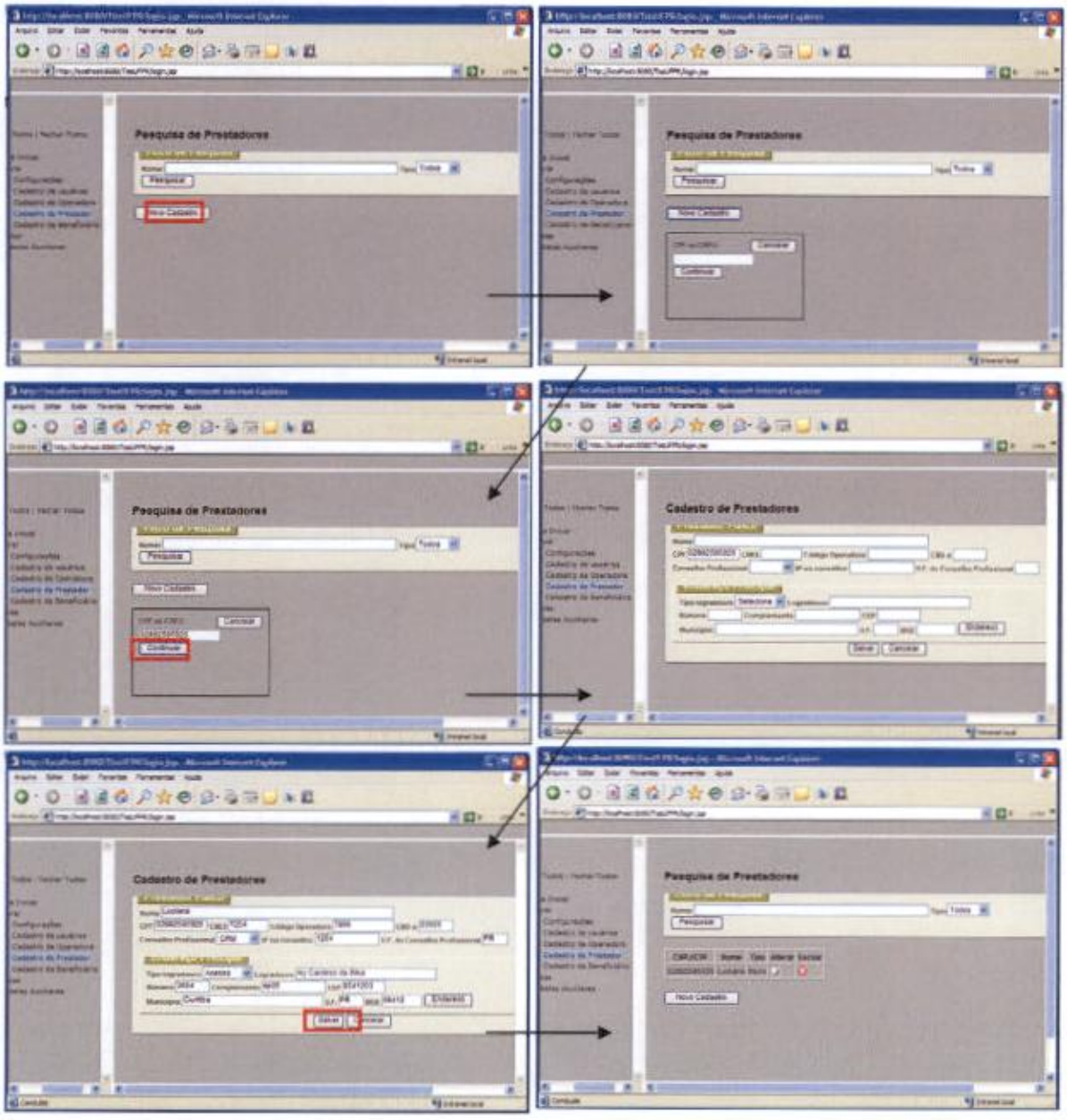
Na tela de pesquisa o usuário digita o nome inicial ou a letra com a qual inicia o nome a procurar e/ou seleciona a opção de filtro para fisico ou jurídico, e clica no botão "Pesquisar". O sistema listará todos os nomes que atendam os parâmetros desejados. Na lista gerada, o usuário clica no ícone excluir ( ) referente ao cadastro que deseja excluir. Aparecerá um alerta solicitando a confirmação da exclusão, o usuário clicando no botão "OK", confirmando a exclusão.

Cancelar

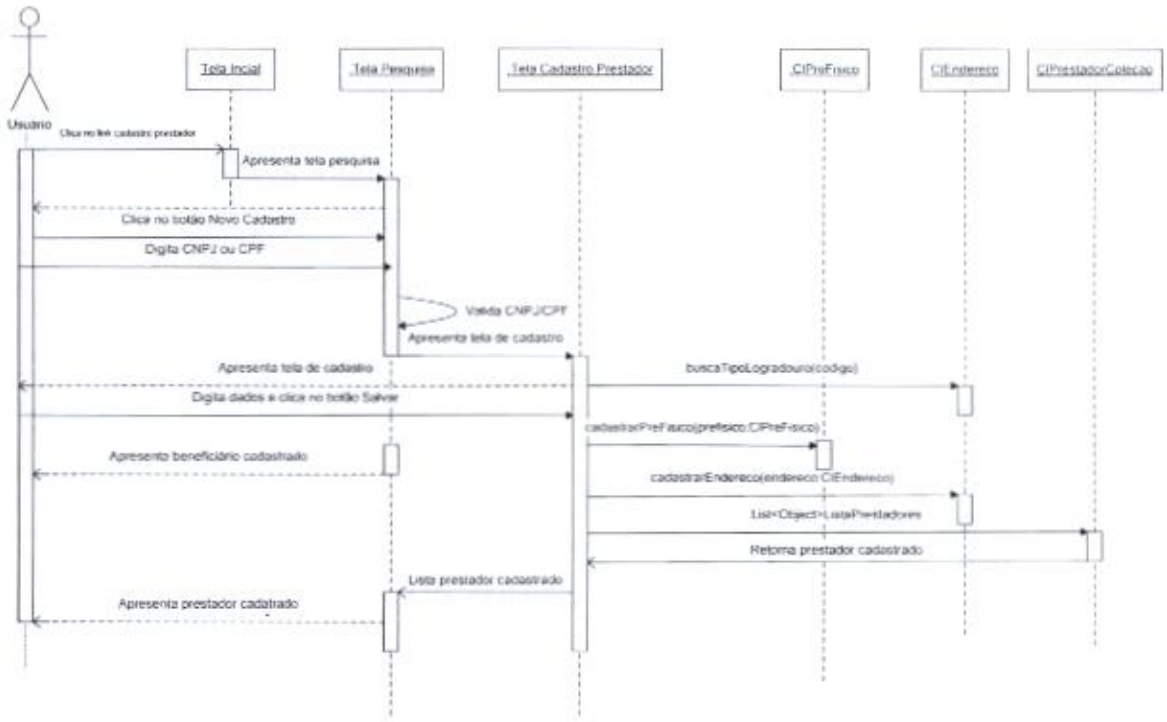
Se o usuário clicar no botão "Cancelar", o sistema interromperá o cadastro.

17.2 Prestador físico

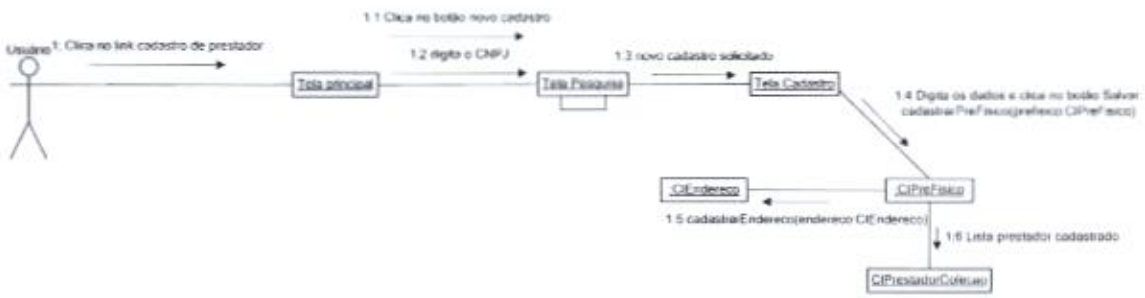
17.2.1 Diagrama de telas – Cadastrar



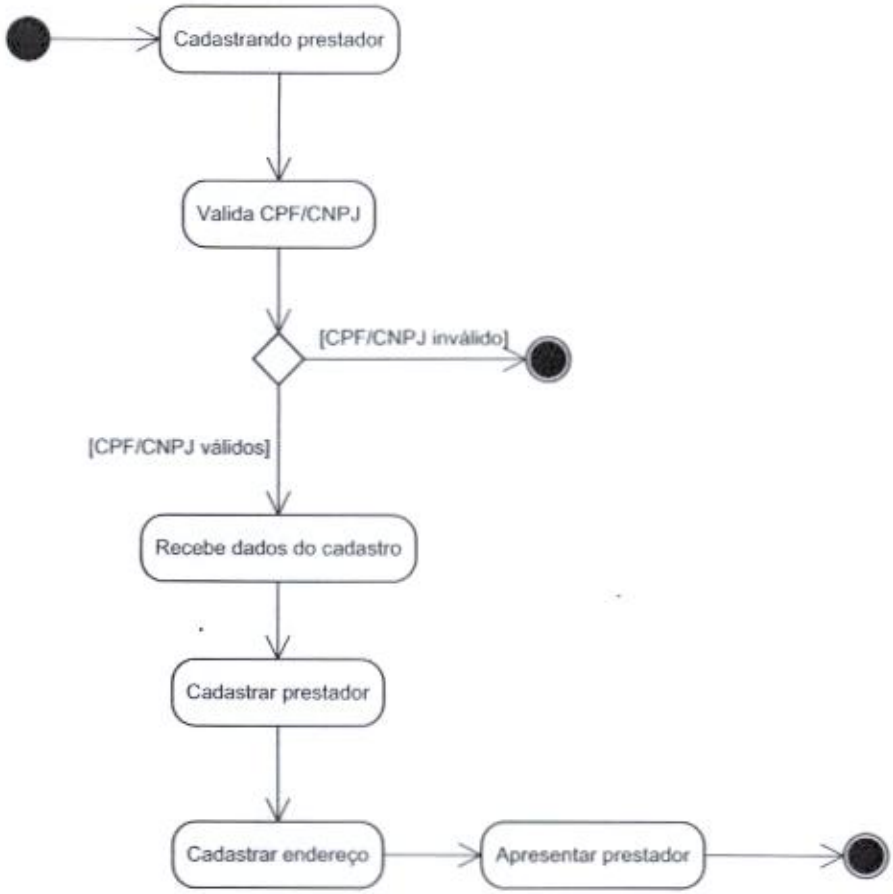
17.2.2 Diagrama de seqüência – Cadastrar



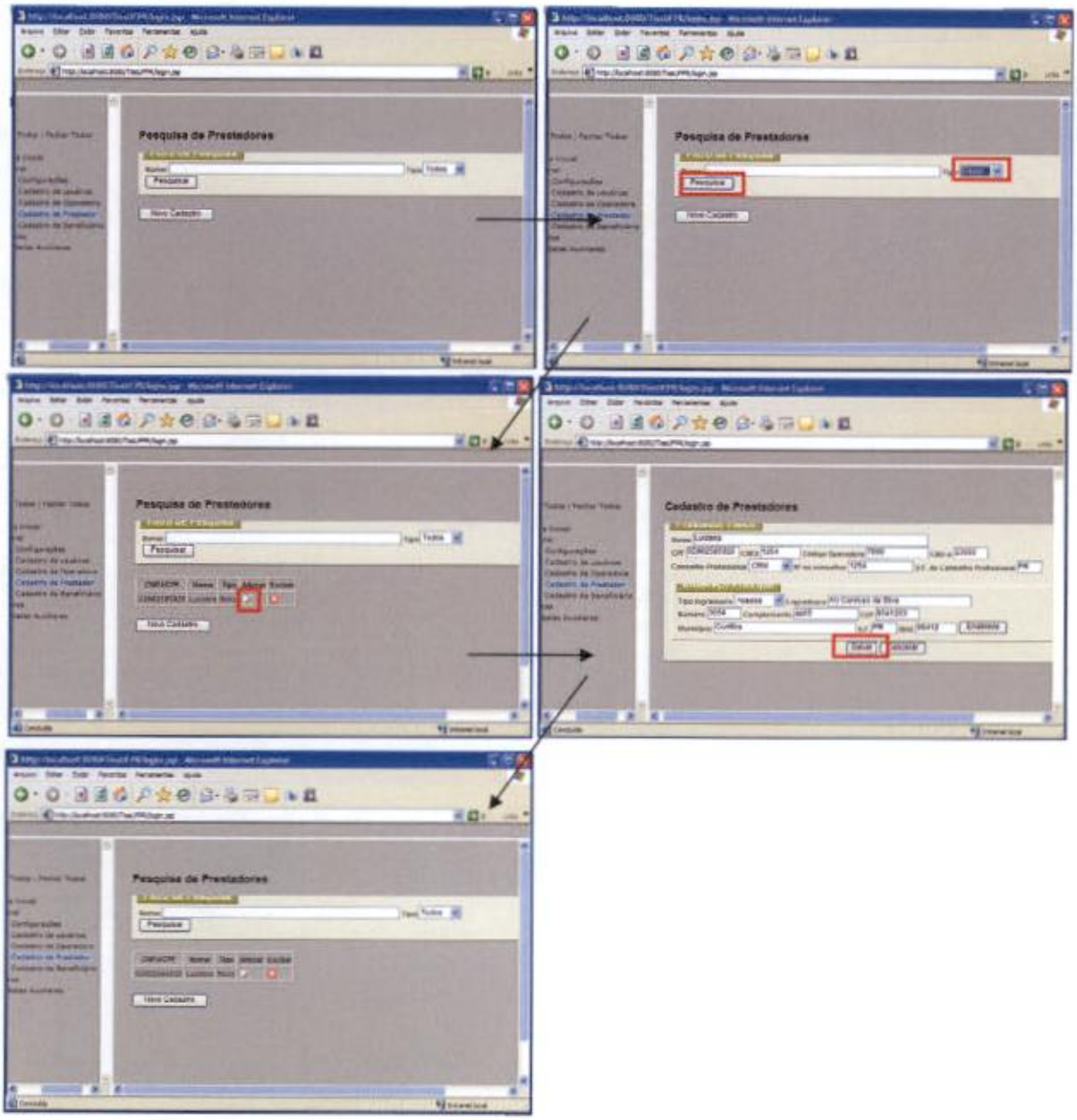
17.2.3 Diagrama de comunicação – Cadastrar



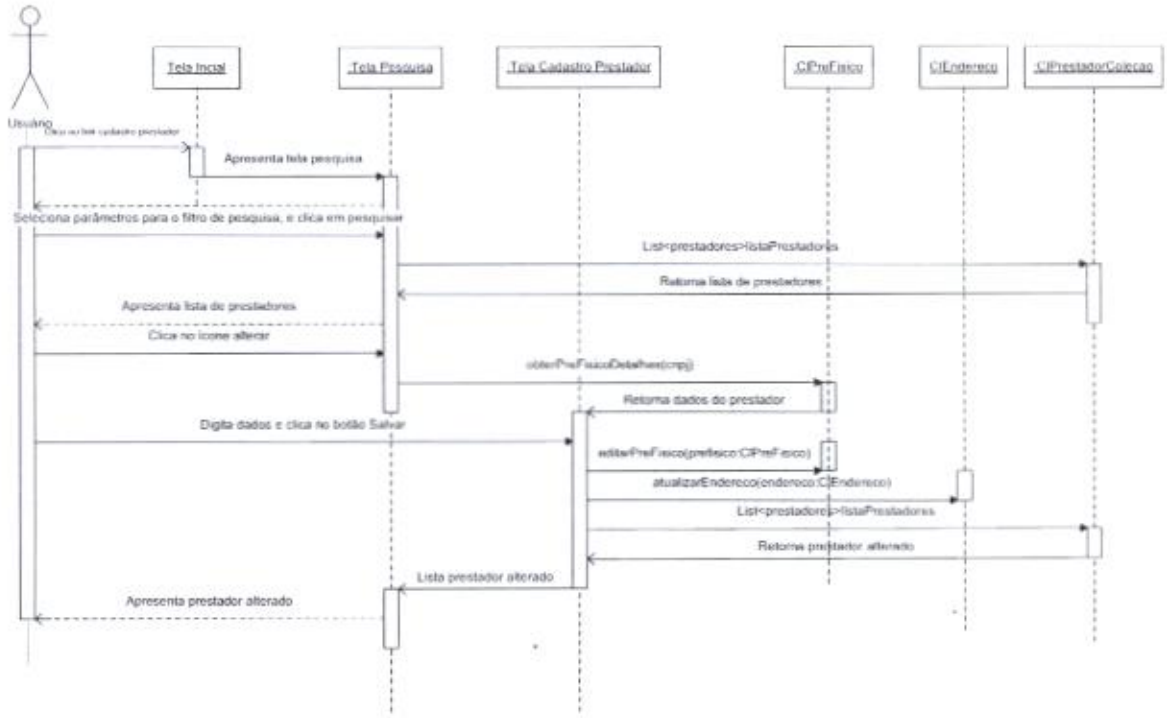
17.2.4 Diagrama de atividades – Cadastrar



17.2.5 Diagrama de telas – Alterar

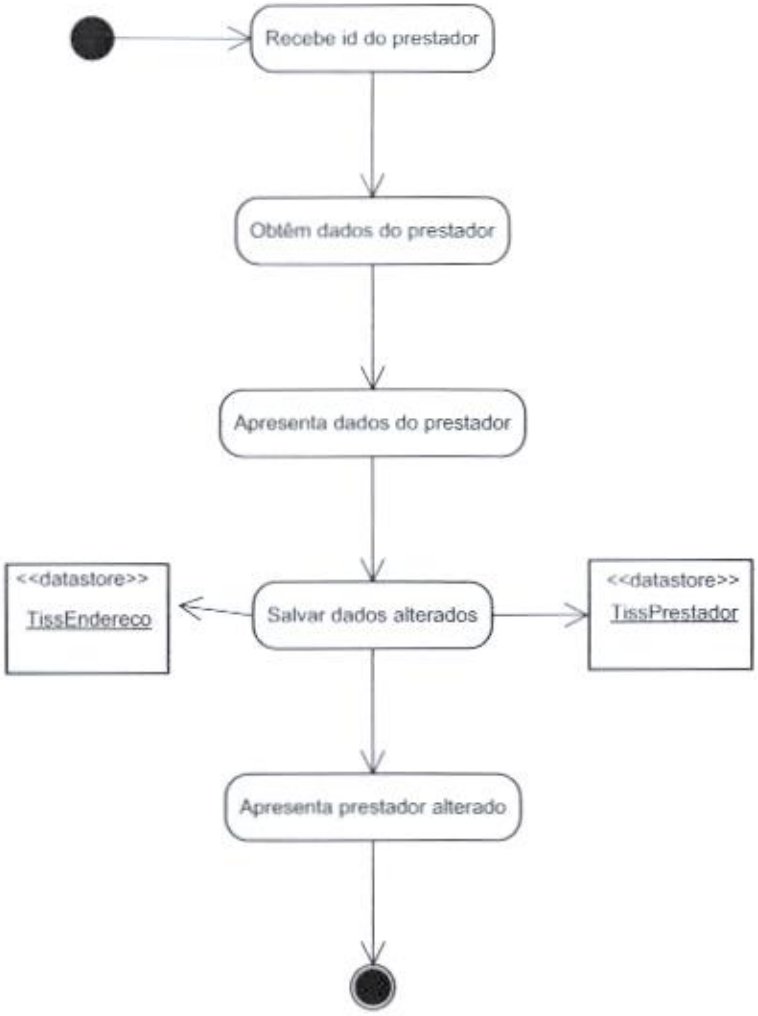


17.2.6 Diagrama de seqüência – Alterar



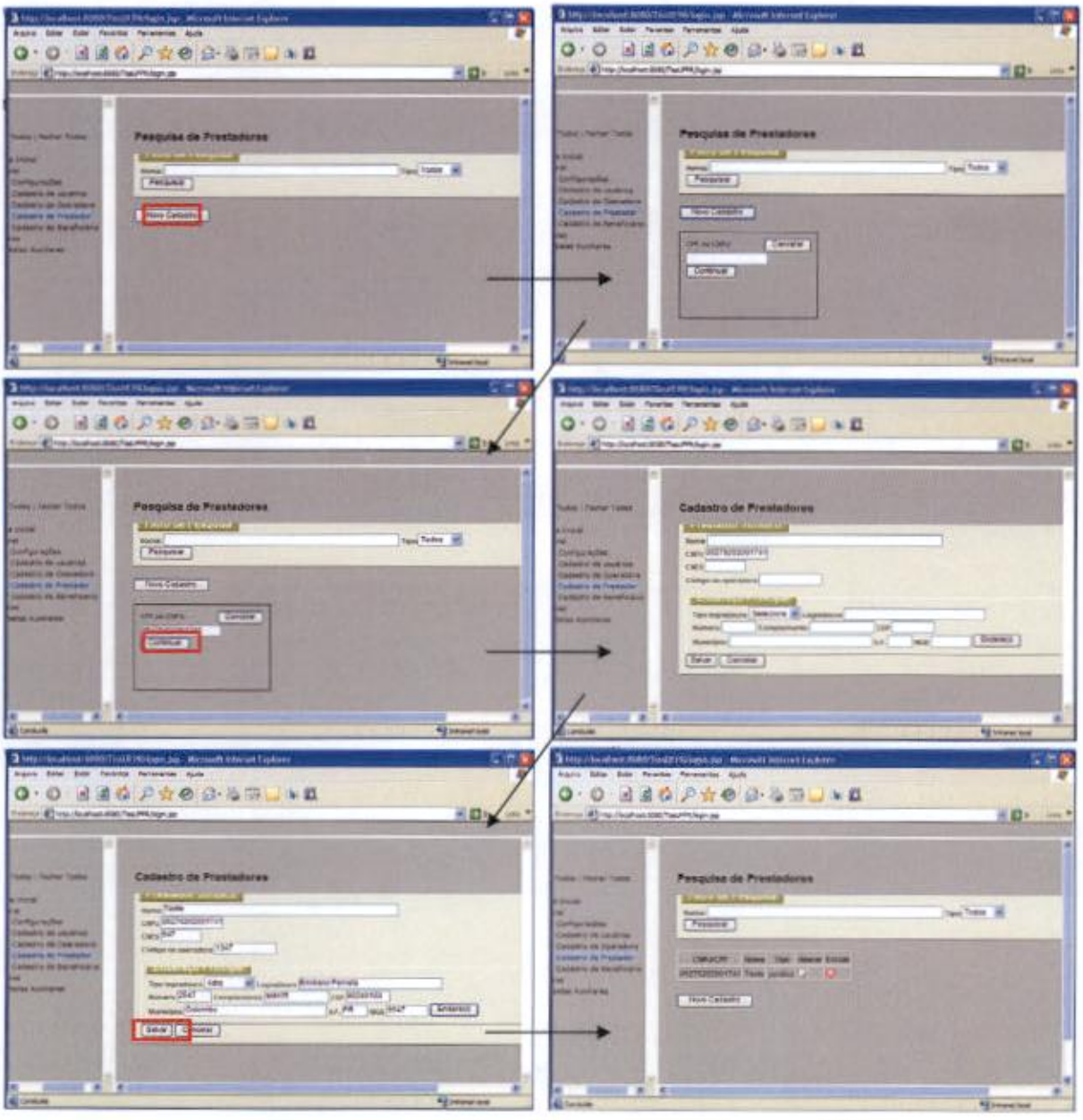
17.2.7 Diagrama de comunicação – Alterar

17.2.8 Diagrama de atividades – Alterar

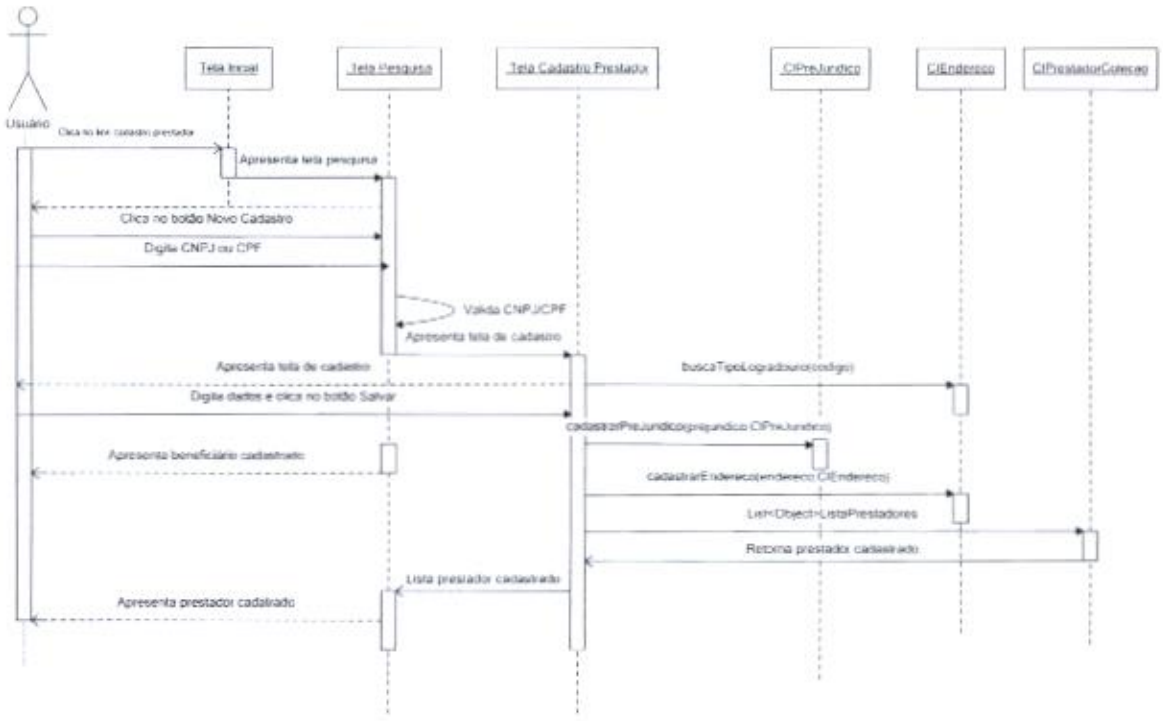


17.3 Prestador jurídico

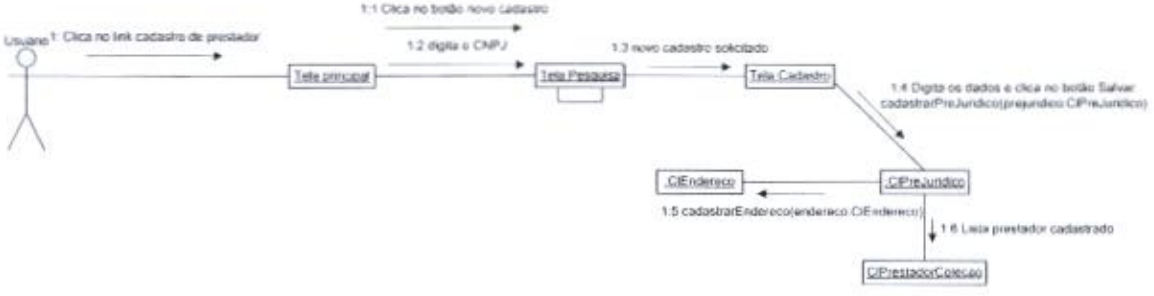
17.3.1 Diagrama de telas – Cadastrar



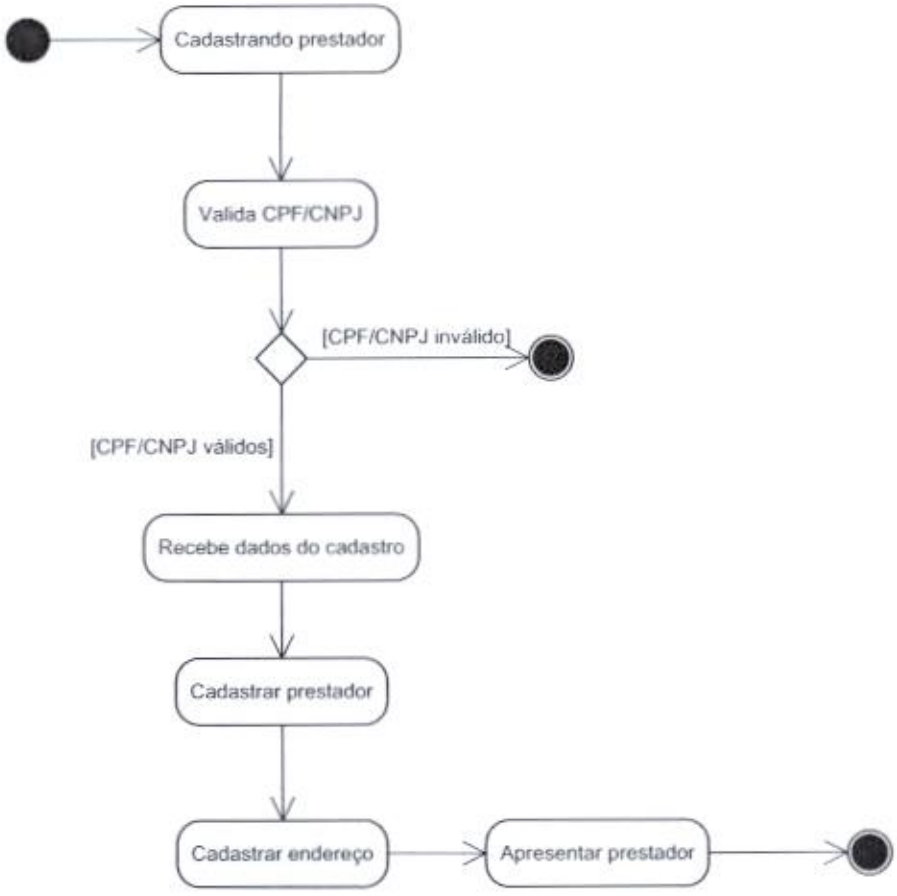
17.3.2 Diagrama de seqüência – Cadastrar



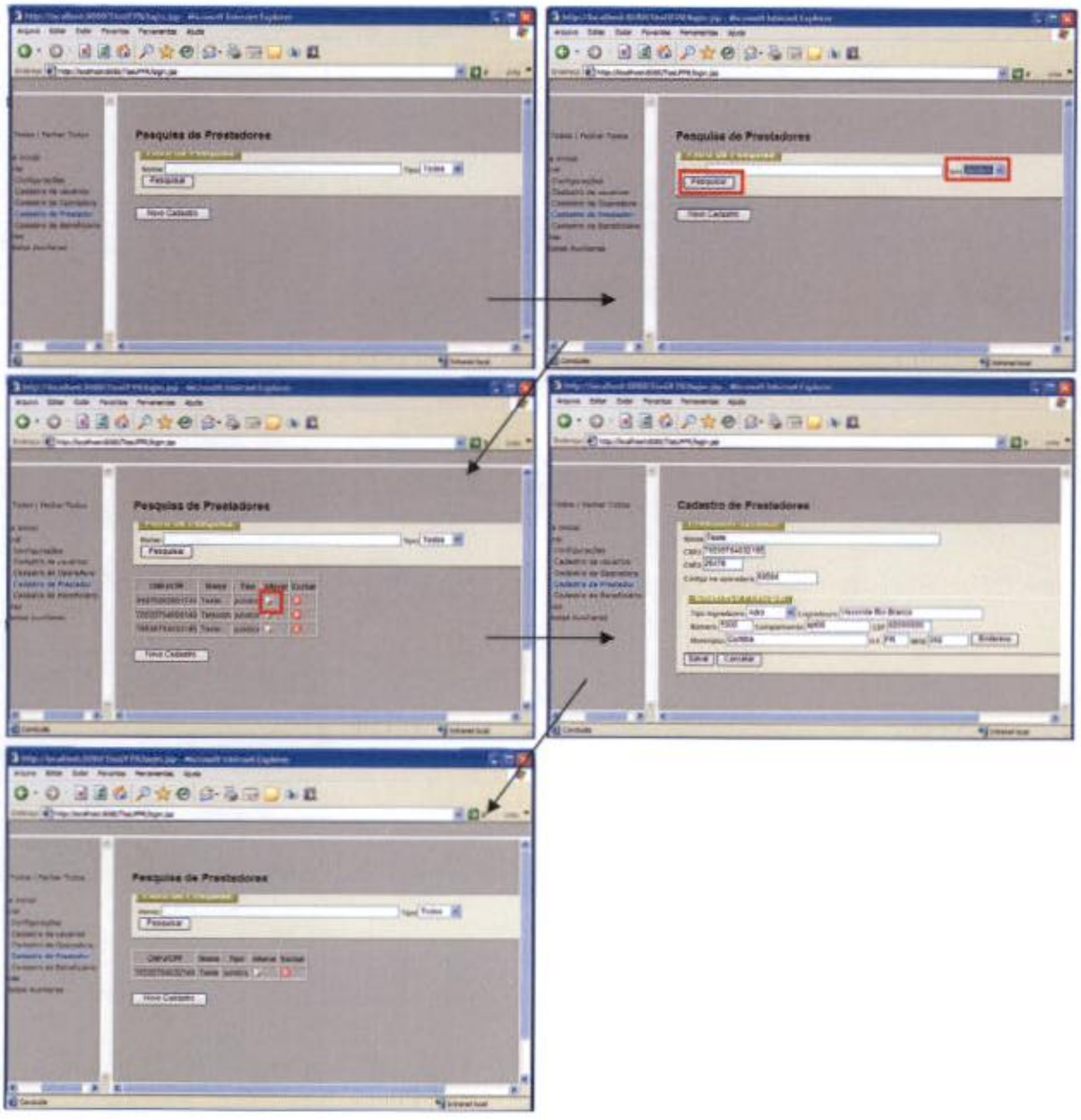
17.3.3 Diagrama de comunicação – Cadastrar



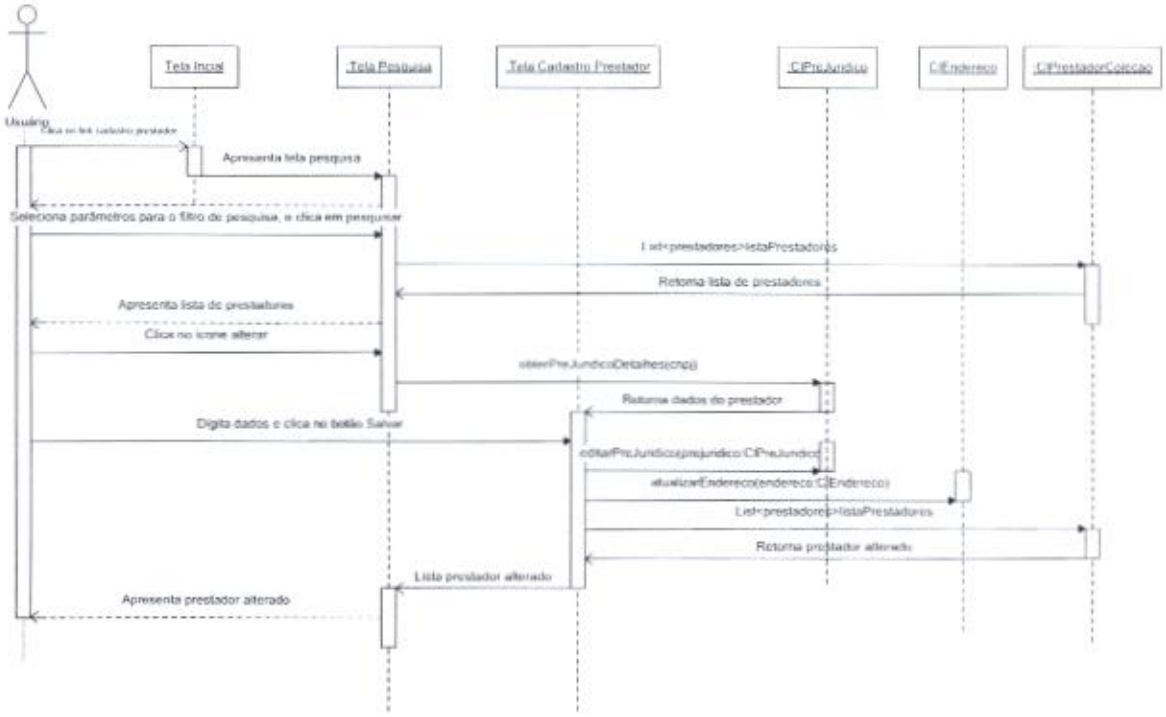
17.3.4 Diagrama de atividades – Cadastrar



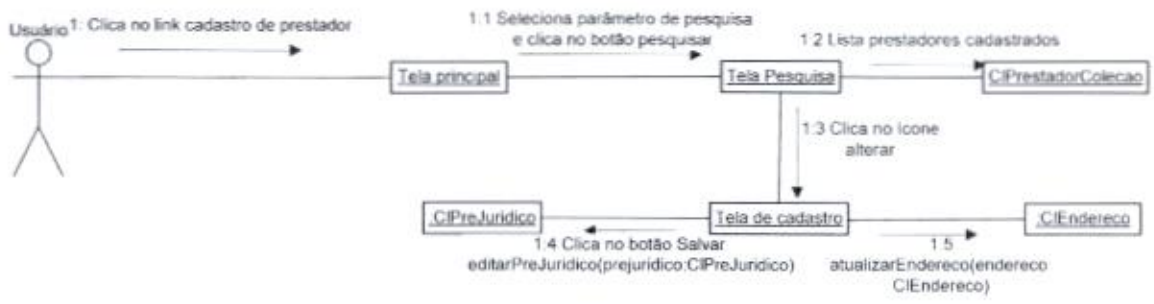
17.3.5 Diagrama de telas – Alterar



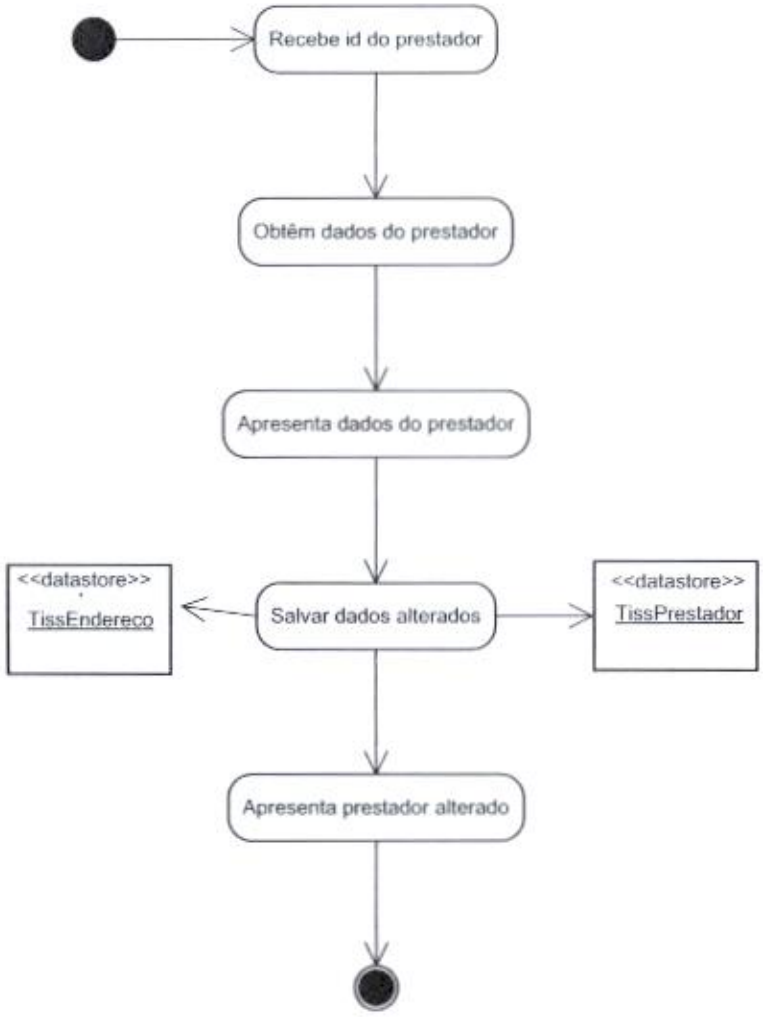
17.3.6 Diagrama de seqüência – Alterar



17.3.7 Diagrama de comunicação – Alterar

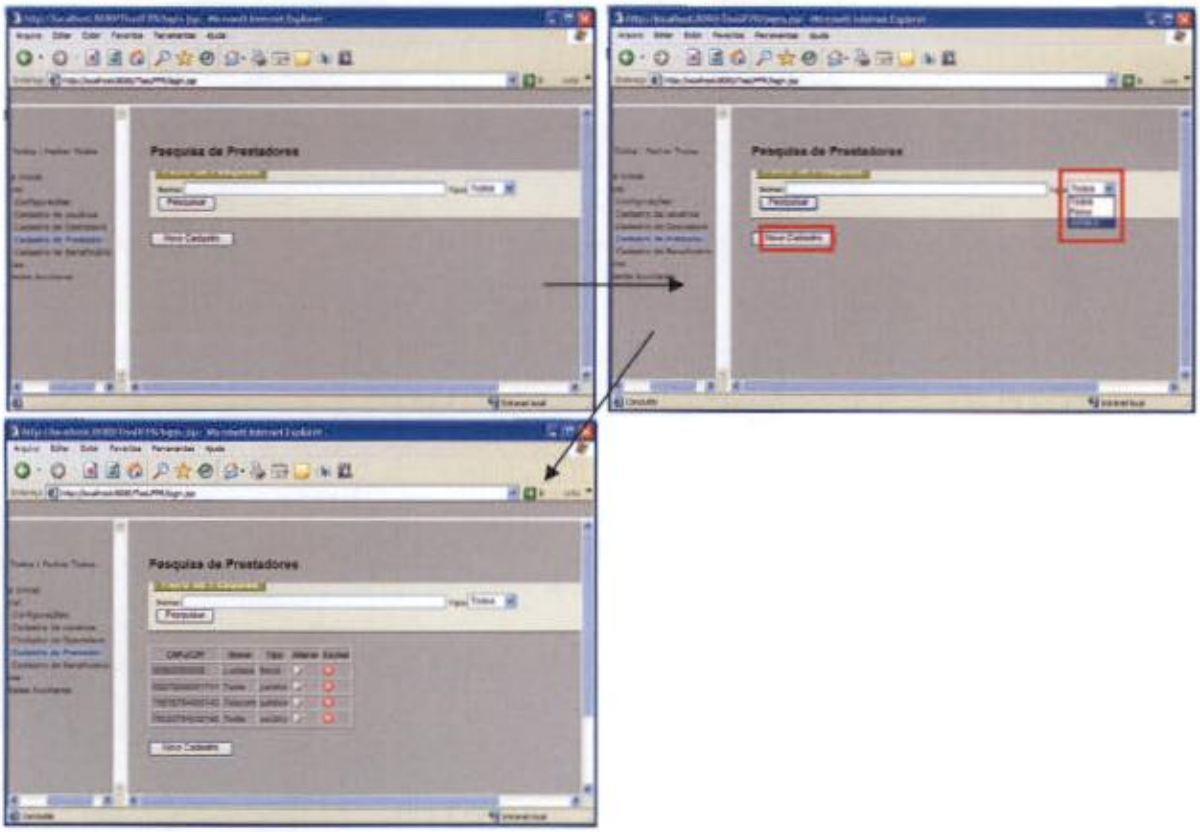


17.3.8 Diagrama de atividades – Alterar

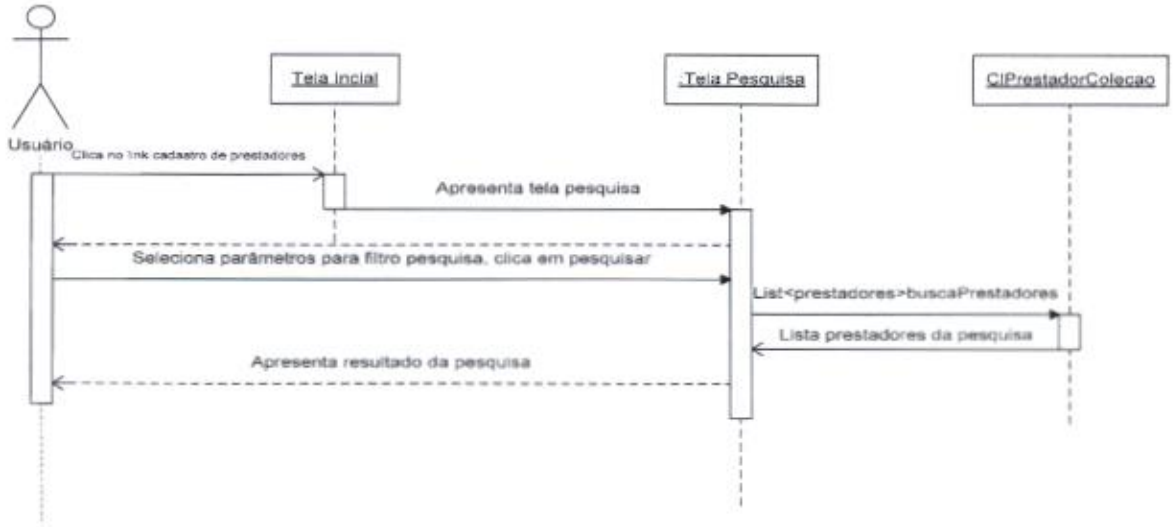


As funções de pesquisa e exclusão seguem os mesmos diagramas tanto para prestador físico quanto para prestador jurídico.

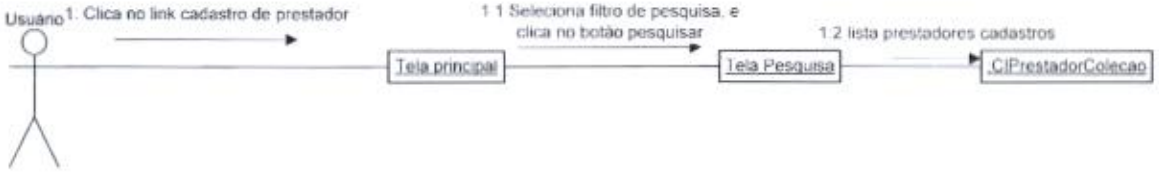
17.4 Diagrama de telas – Pesquisar



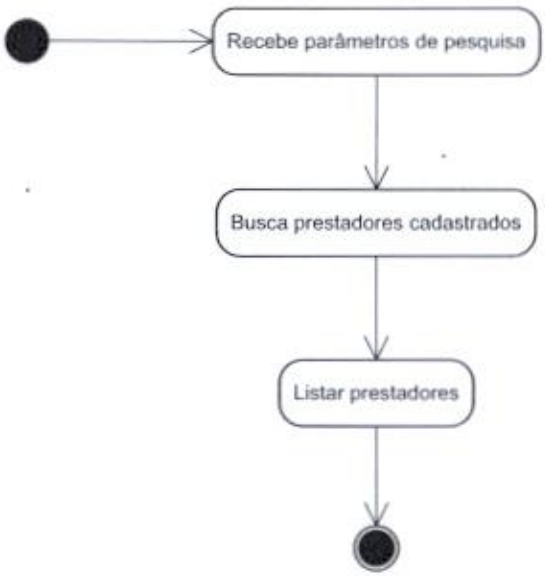
17.5 Diagrama de seqüência – Pesquisar



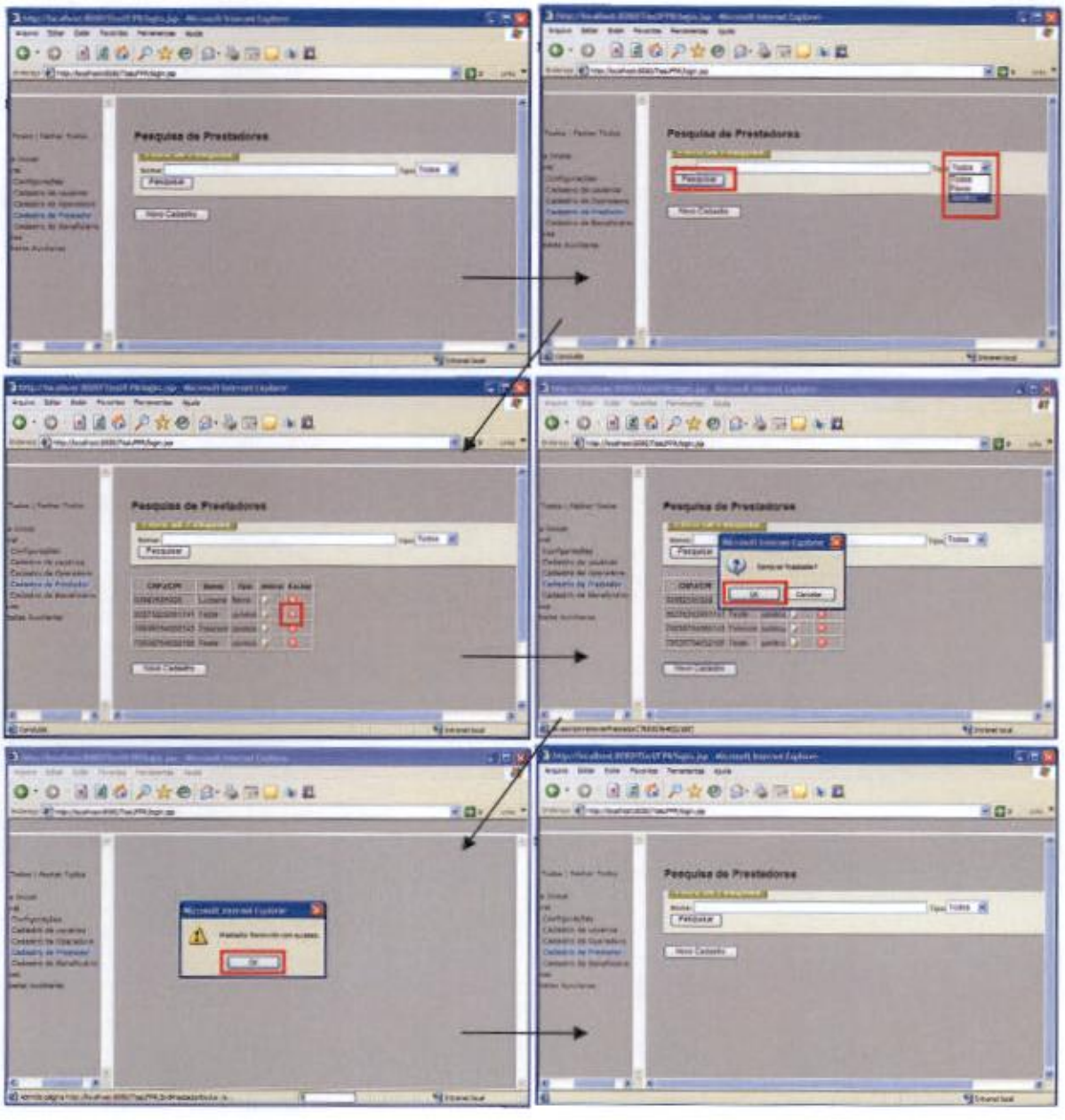
17.6 Diagrama de comunicação – Pesquisar



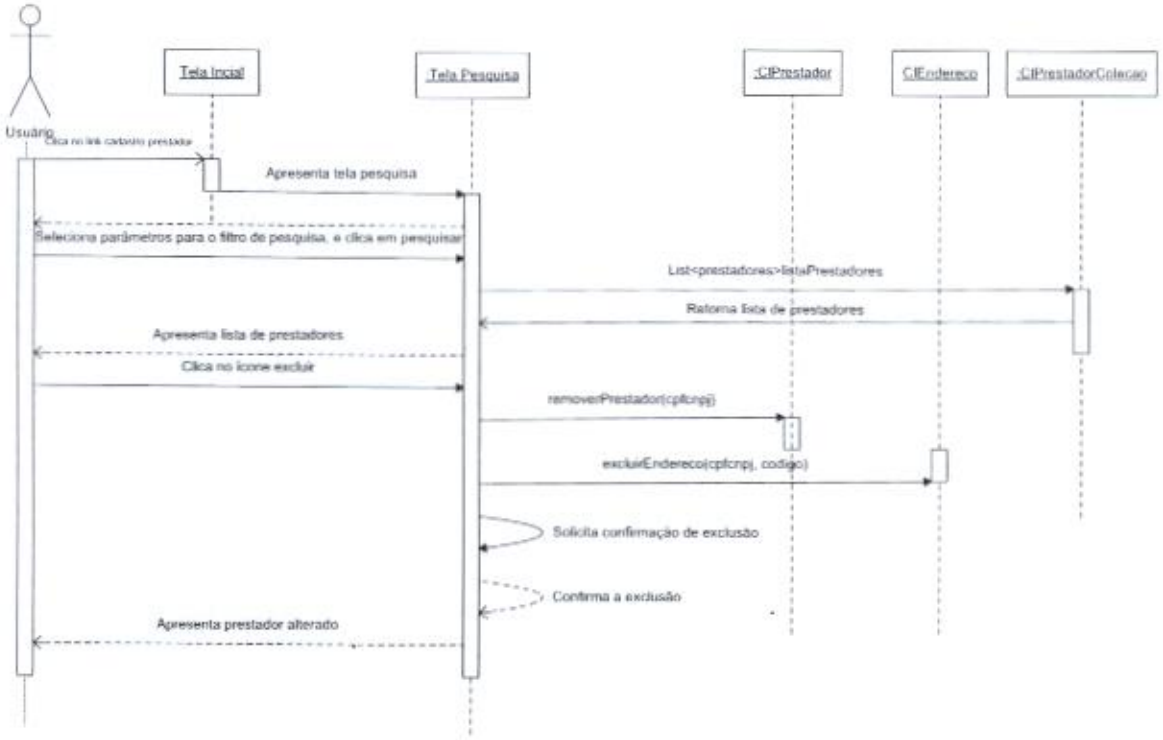
17.7 Diagrama de atividades – Pesquisar



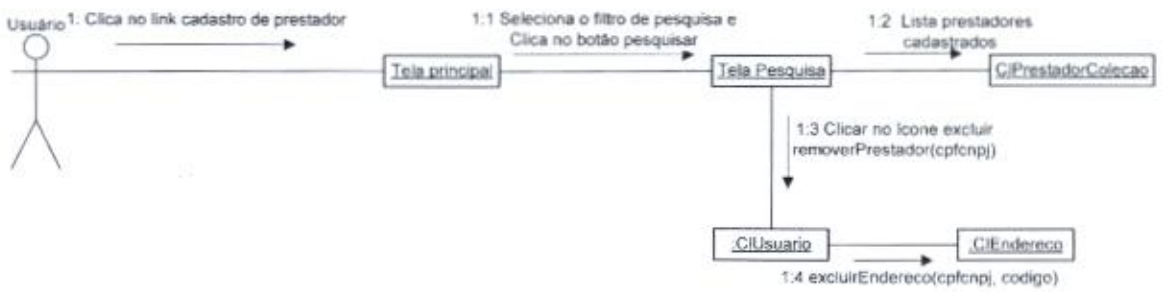
17.8 Diagrama de telas – Excluir



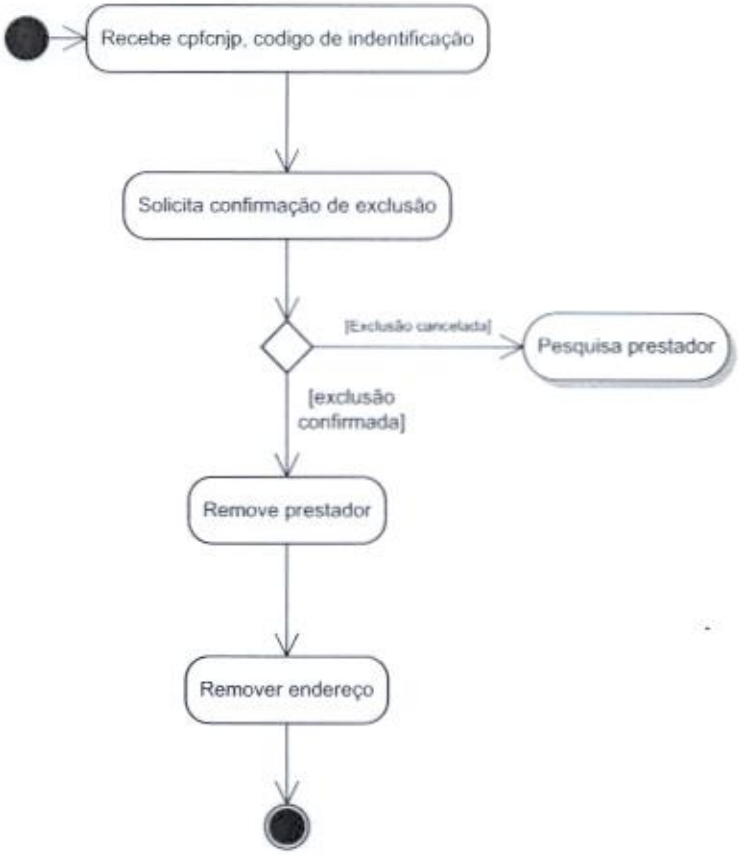
17.9 Diagrama de seqüência – Excluir



17.10 Diagrama de comunicação – Excluir



17.11 Diagrama de atividades - Excluir



18. GERAR GUIAS

18.1 Eventos do caso de uso

Pré-requisito: Já possuir cadastrados de operadoras, beneficiários e prestadores no sistema.

Fluxo principal

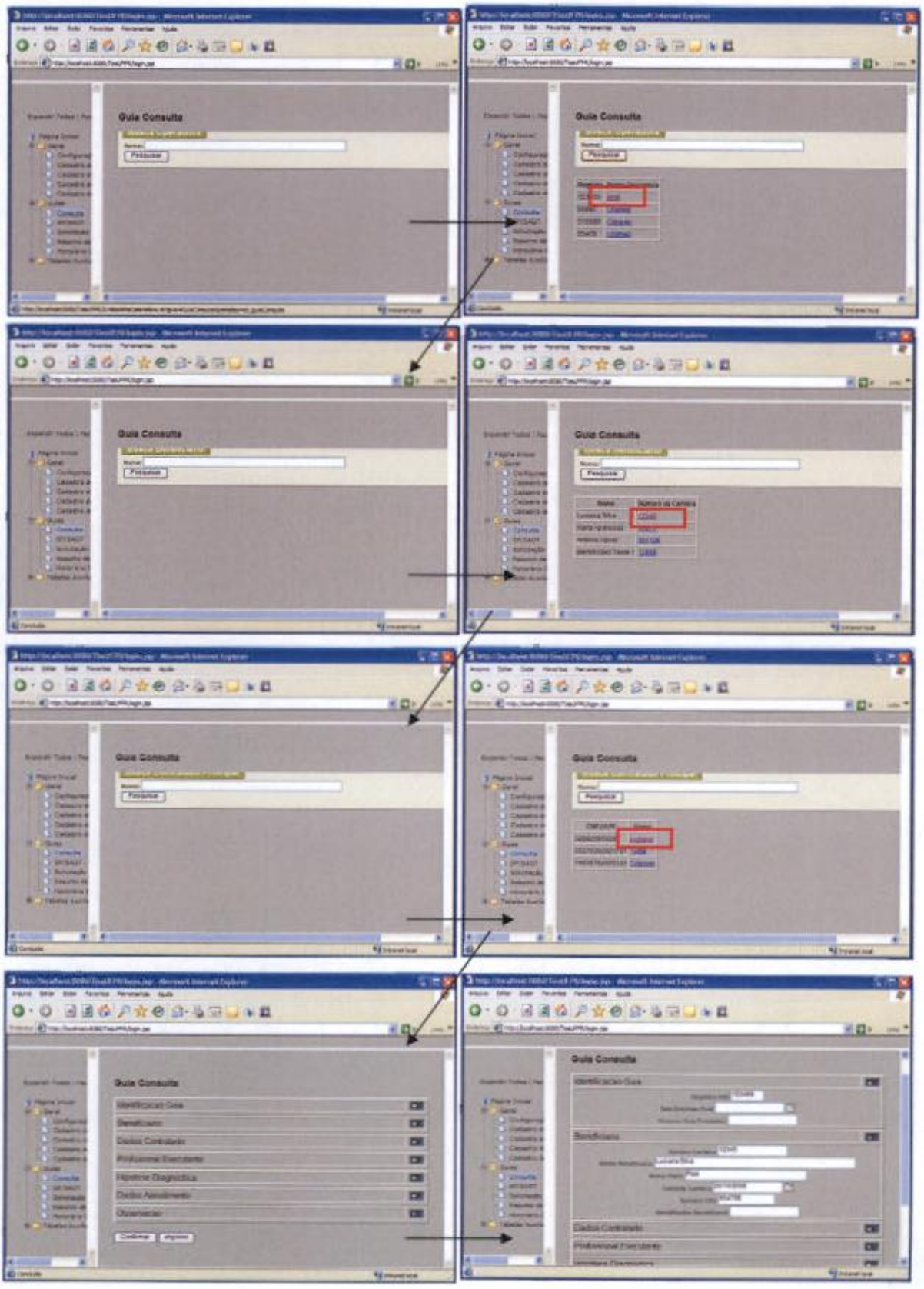
O usuário clica no link da guia desejada e o sistema abrirá a tela para pesquisar a operadora, após a seleção da operadora, abrirá a tela de seleção de beneficiário. O usuário escolhe o beneficiário para o qual será feita a guia e, após a seleção, o sistema abrirá a tela para a escolha do prestador, assim que o usuário selecionar o prestador o sistema gerará a guia em questão já com os campos referentes às seleções anteriores pré-preenchidos.

Fluxos alternativos

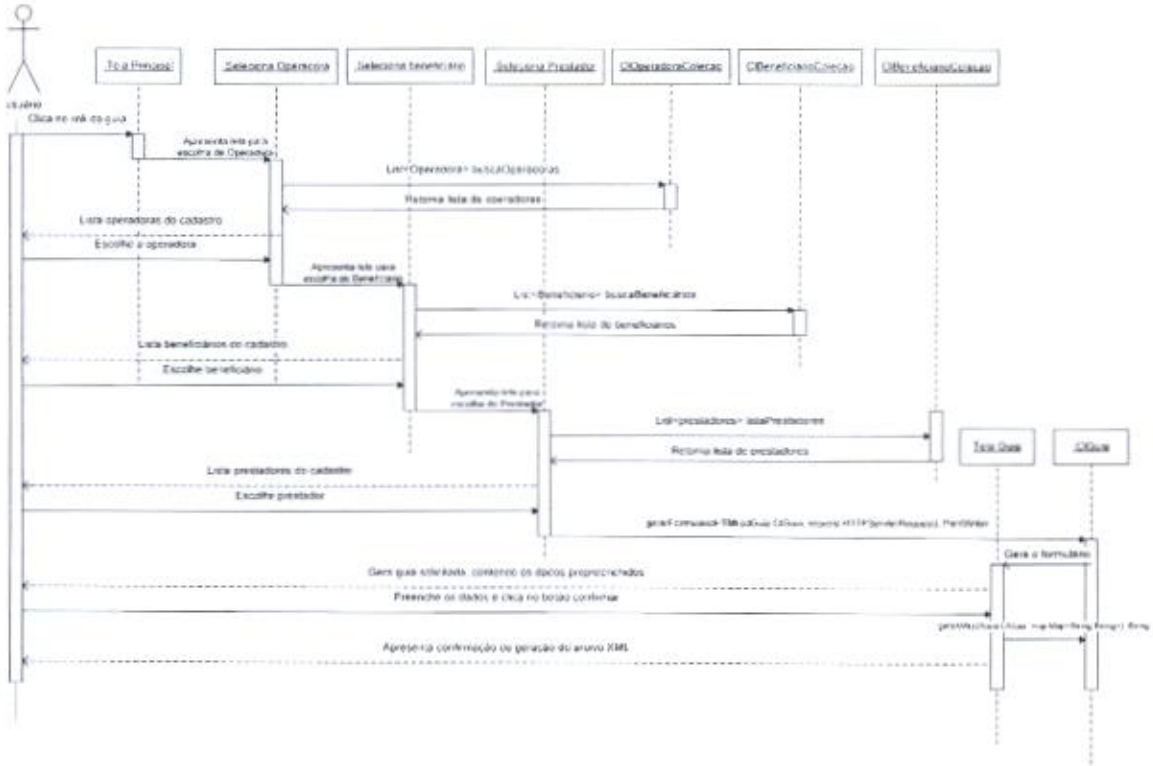
Para esse caso de uso, não possuímos fluxos alternativos.

As gerações das demais guias são sob o mesmo fluxo.

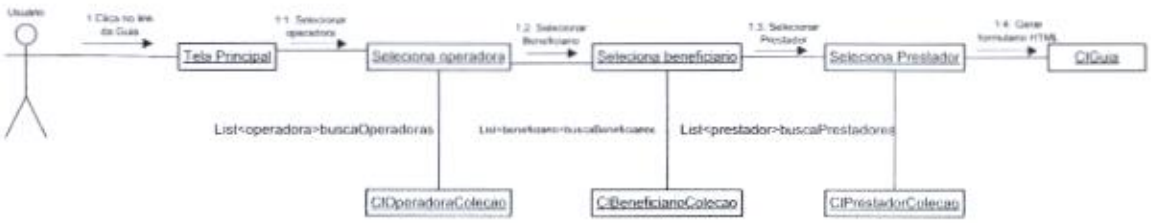
18.2 Diagrama de telas



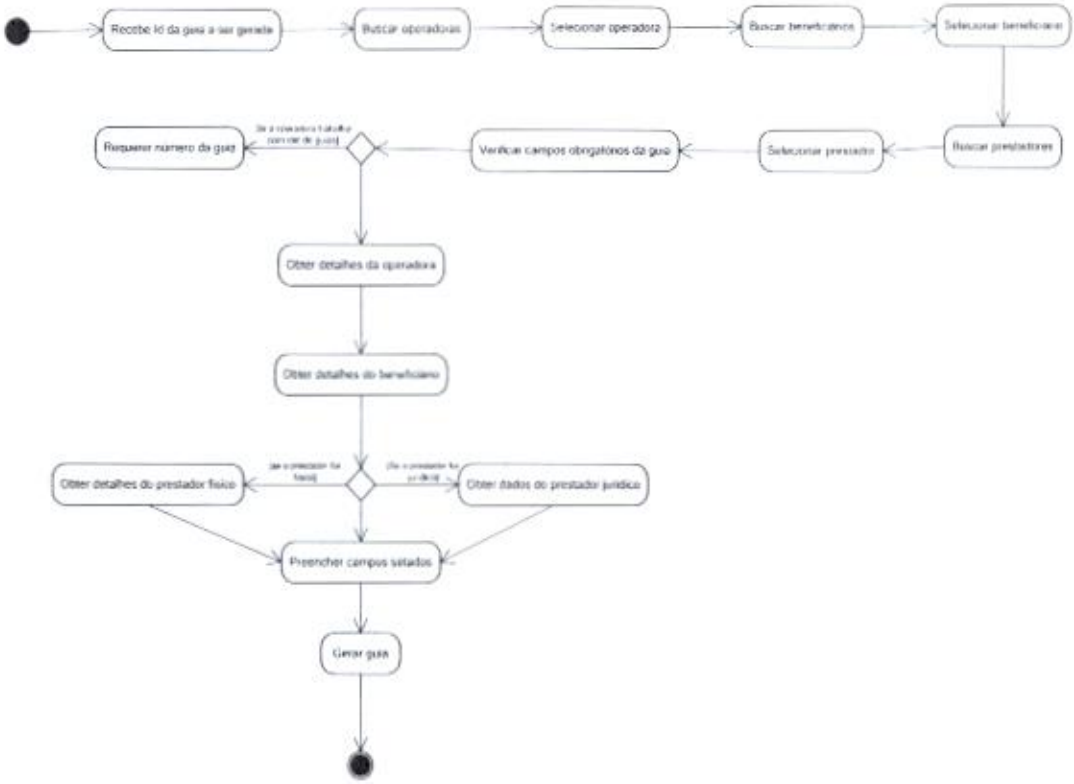
18.3 Diagrama de seqüência



18.4 Diagrama de comunicação



18.5 Diagrama de atividades



19. GERAR XML DAS GUIAS

19.1 Eventos do caso de uso

Pré-requisito: ter realizado o preenchimento dos campos obrigatórios da guia.

Fluxo principal

O usuário clica no link da guia desejada e o sistema abrirá a tela para pesquisar a operadora, após a seleção da operadora, abrirá a tela de seleção de beneficiário. O usuário escolhe o beneficiário para o qual será feita a guia e, após a seleção, o sistema abrirá a tela para a escolha do prestador, assim que o usuário selecionar o prestador o sistema gerará a guia em questão já com os campos referentes às seleções anteriores pré-preenchidos.

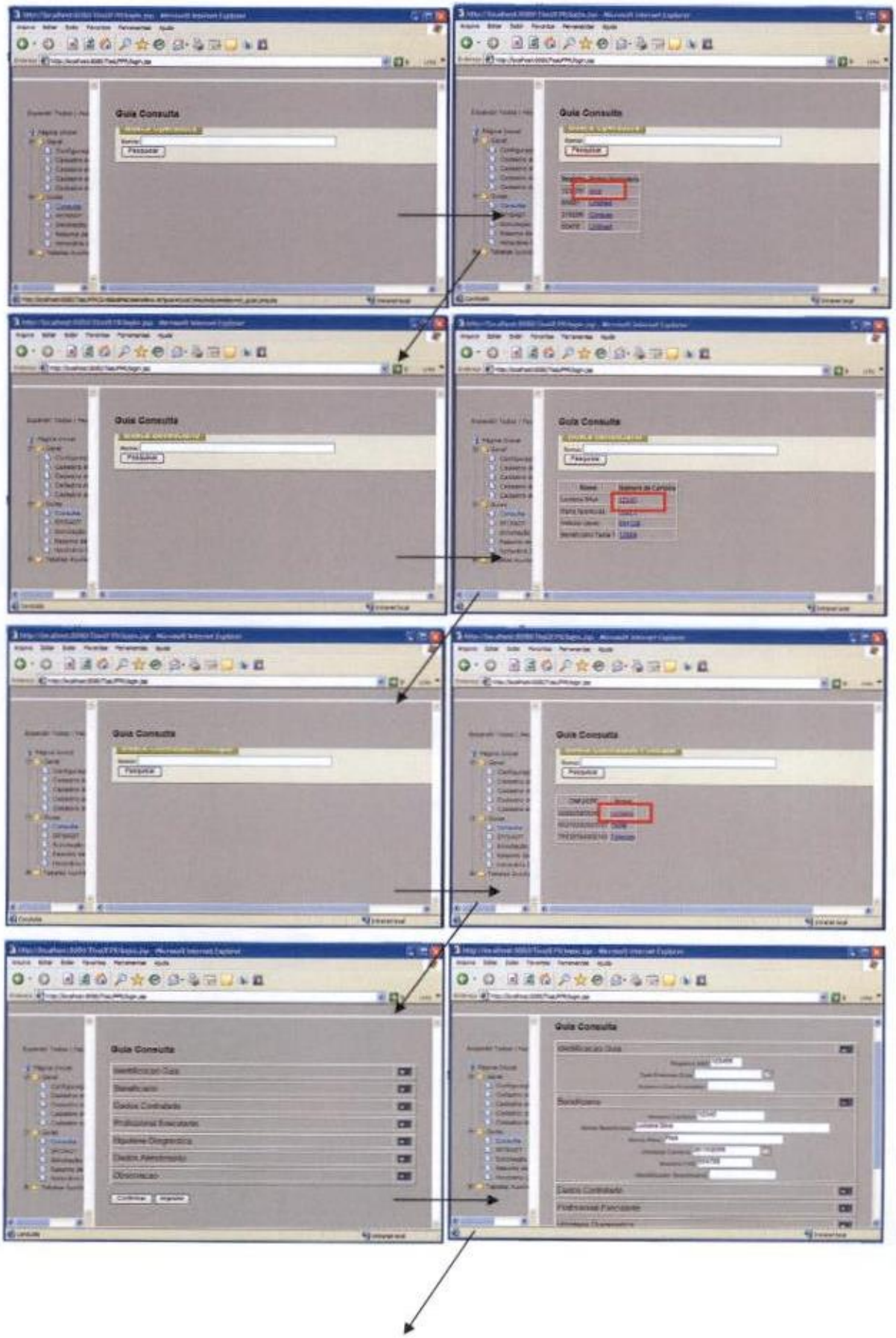
Após o usuário preencher os demais campos solicitados na guia, ele clica no botão confirma e o sistema irá gerar o arquivo XML referente a essa guia, arquivando-o na pasta raiz do computar.

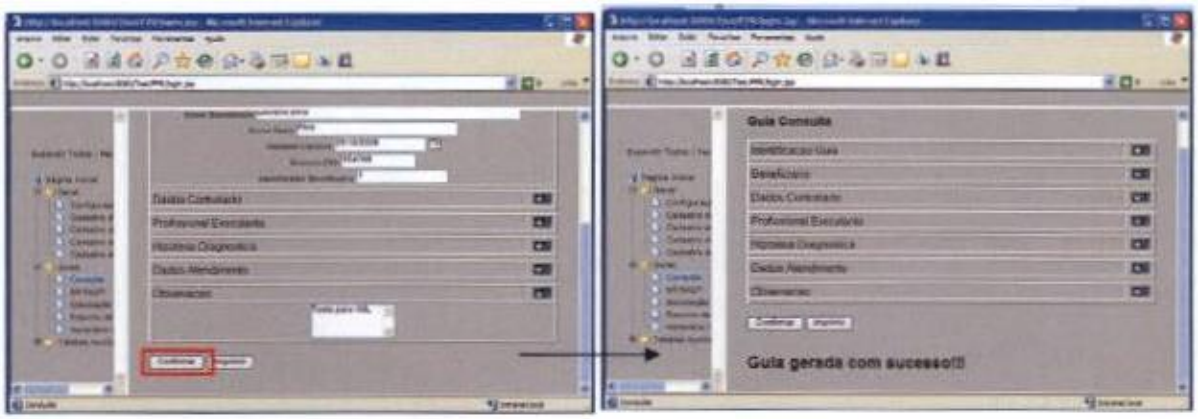
Fluxos alternativos

Para esse caso de uso, não possuímos fluxos alternativos.

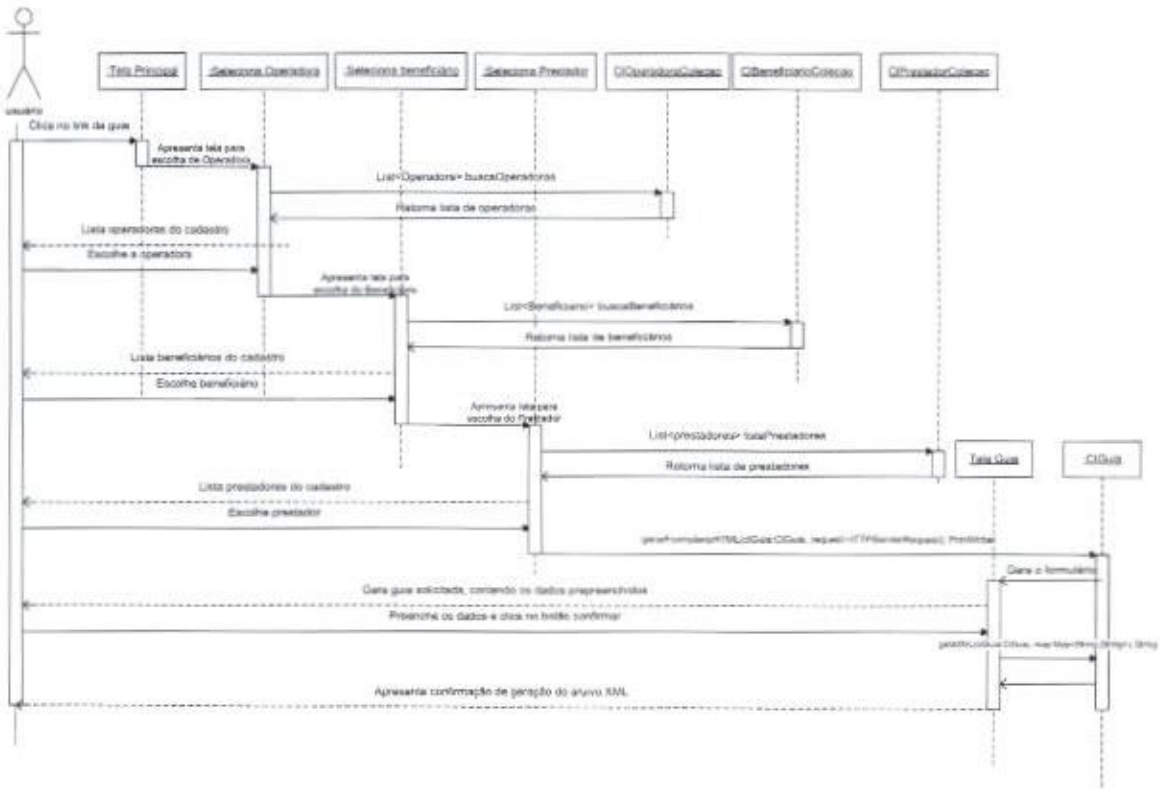
Todas as guias seguem o mesmo caso de uso, para a geração do arquivo XML.

19.2 Diagrama de telas

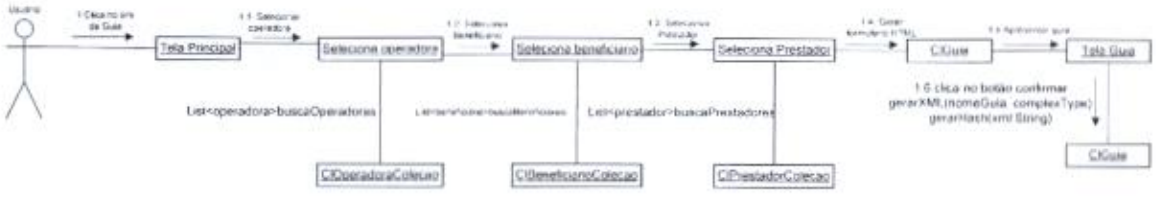




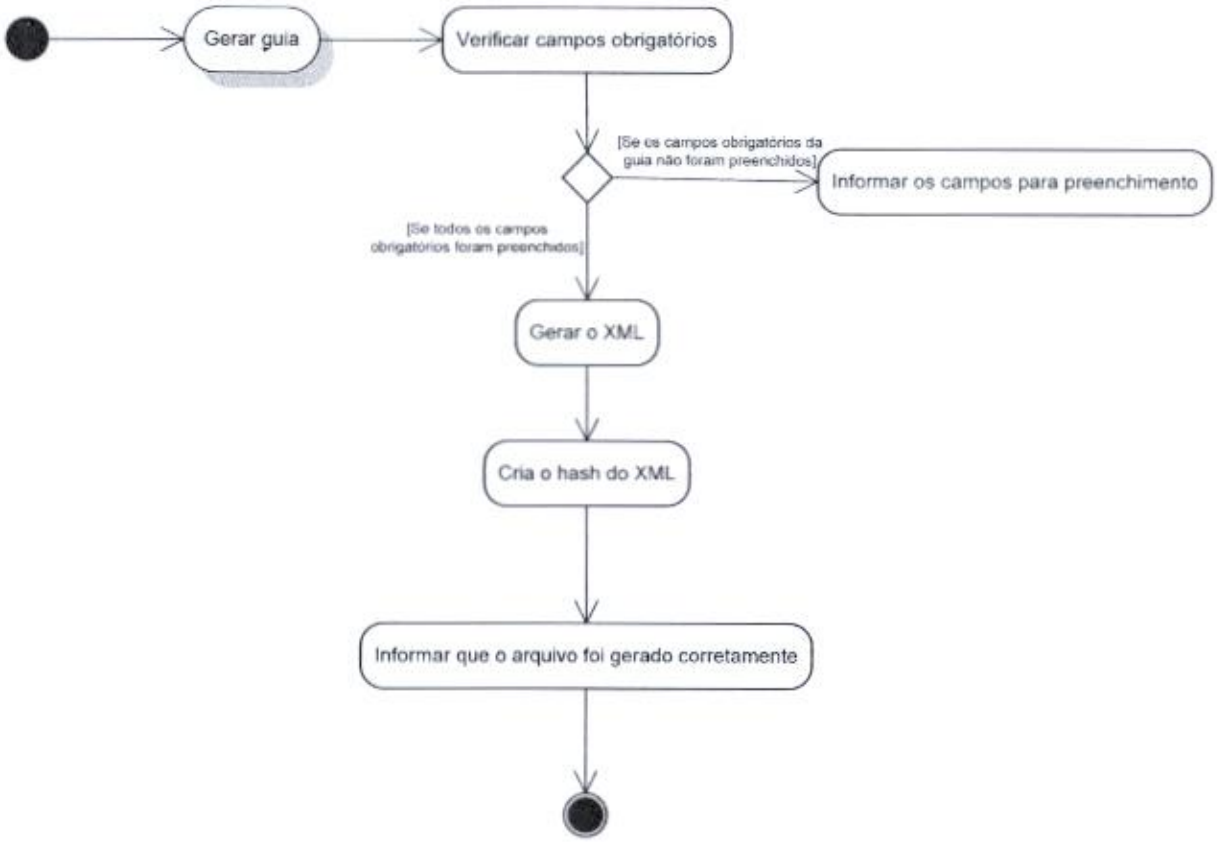
19.3 Diagrama de seqüência



19.4 Diagrama de comunicação



19.5 Diagrama de atividades



20. IMPRIMIR GUIAS

20.1 Eventos do caso de uso

Pré-requisito: não gerar o arquivo XML antes de imprimir a guia.

Fluxo principal

O usuário clica no link da guia desejada e o sistema abrirá a tela para pesquisar a operadora, após a seleção da operadora, abrirá a tela de seleção de beneficiário. O usuário escolhe o beneficiário para o qual será feita a guia e, após a seleção, o sistema abrirá a tela para a escolha do prestador, assim que o usuário selecionar o prestador o sistema gerará a guia em questão já com os campos referentes às seleções anteriores pré-preenchidos.

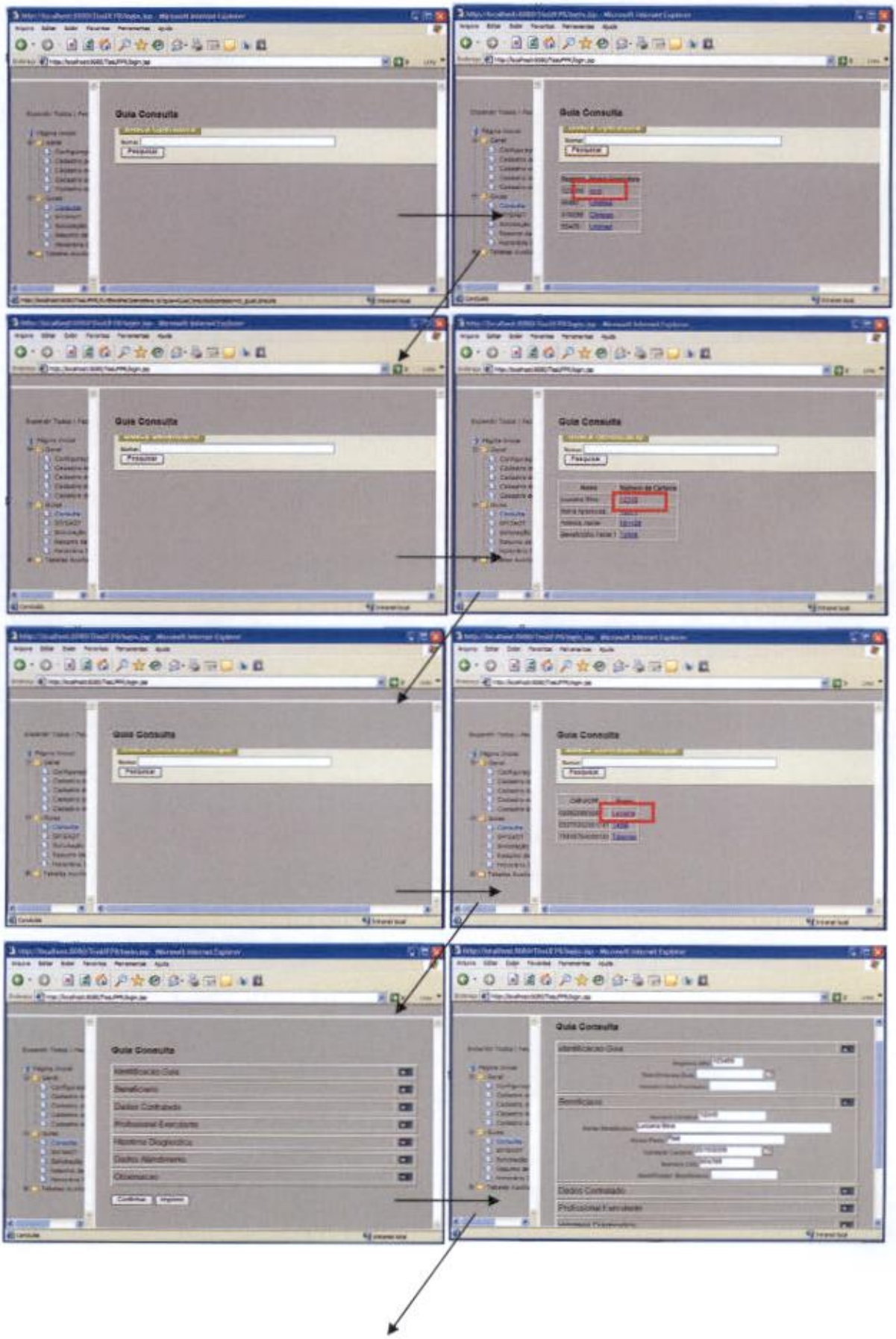
Após o usuário preencher os demais campos solicitados na guia, ele clica no botão imprimir e o sistema abrirá uma nova tela do browser com o documento da guia preenchida permitindo a sua impressão.

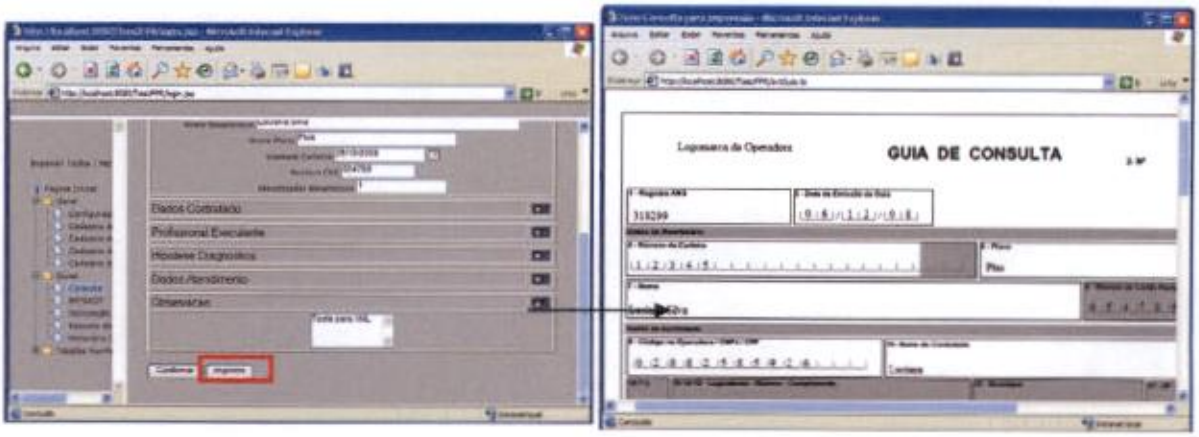
Fluxos alternativos

Caso o usuário queira visualizar uma guia que já tenha sido gerado o arquivo XML, ele deve utilizar a opção de consultar guia.

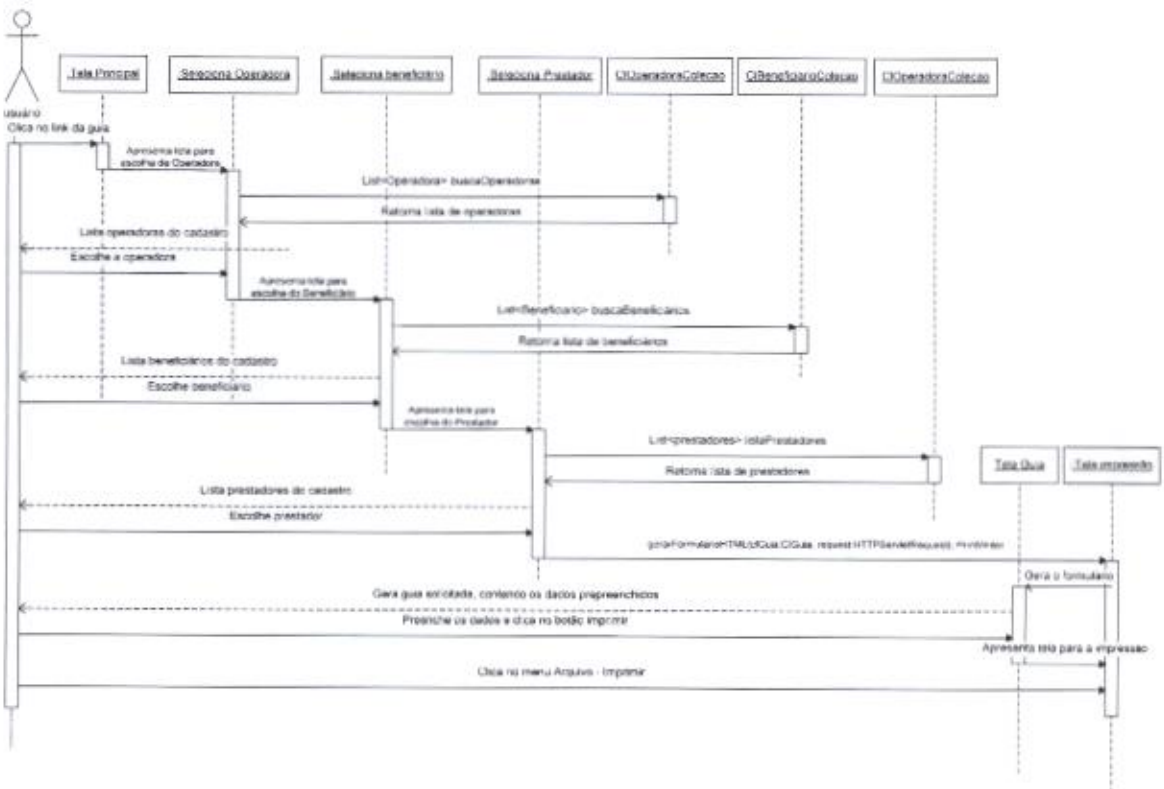
Todas as guias utilizam o mesmo caso de uso.

20.2 Diagrama de telas

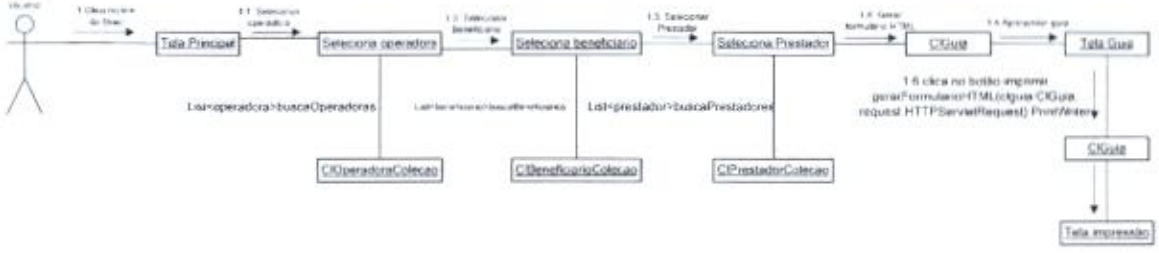




20.3 Diagrama de seqüência



20.4 Diagrama de comunicação



20.5 Diagrama de atividades

21. CONSULTAR GUIA

21.1 Eventos do caso de uso

Pré-requisito: Ter gerado o XML da guia

Fluxo principal

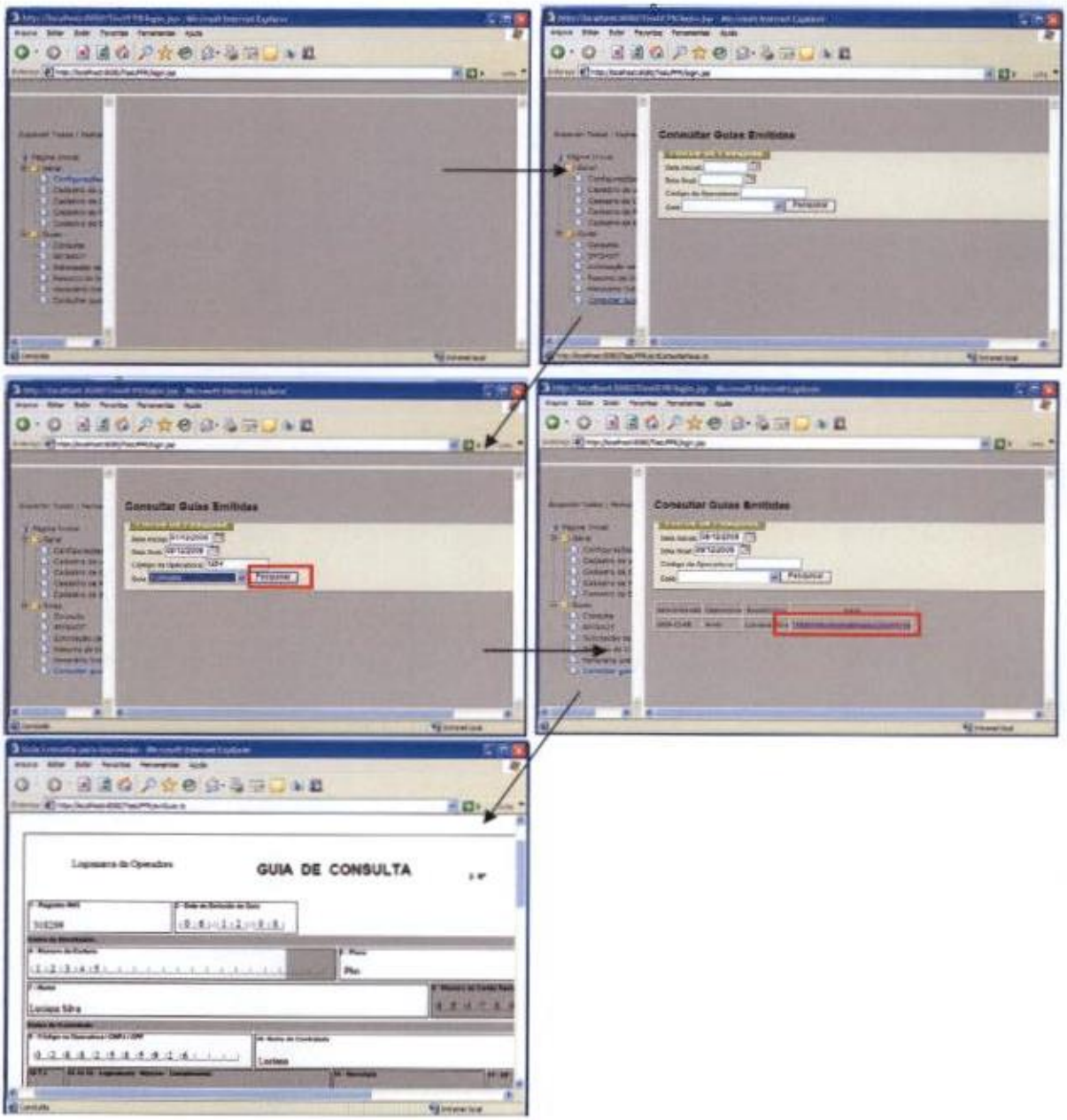
O usuário seleciona a opção consultar guia no menu principal. Abrirá a tela de consulta, onde ele deve selecionar os parâmetros da pesquisa. Após indicar esses parâmetros, ele deve clicar no botão pesquisar e o sistema demonstrará os arquivos salvos, que atendam a especificação da procura.

Com os dados em tela, o usuário pode clicar no código hash do arquivo, para visualizar a guia e imprimí-la.

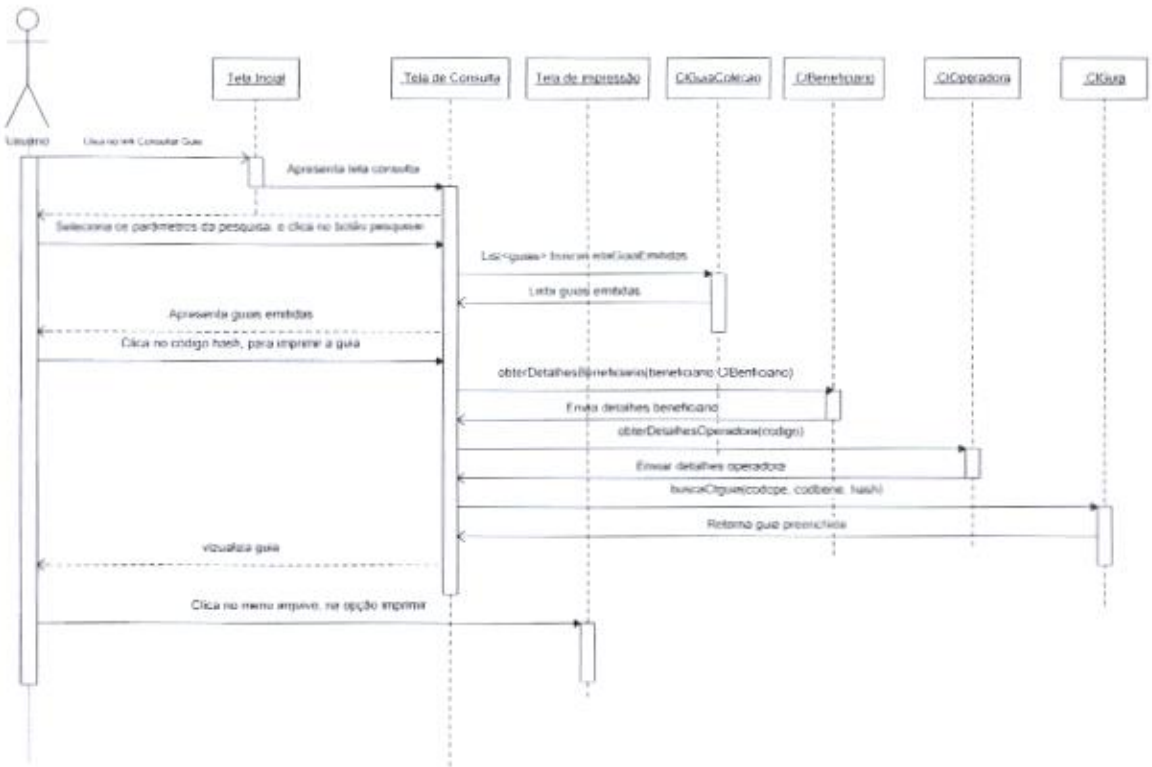
Fluxo alternativo

Para esse caso de uso, não possuímos fluxo alternativo.

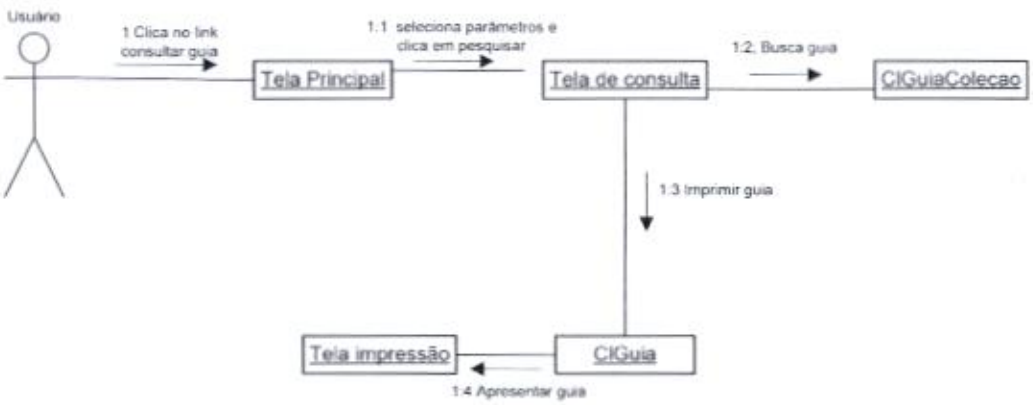
21.2 Diagrama de telas



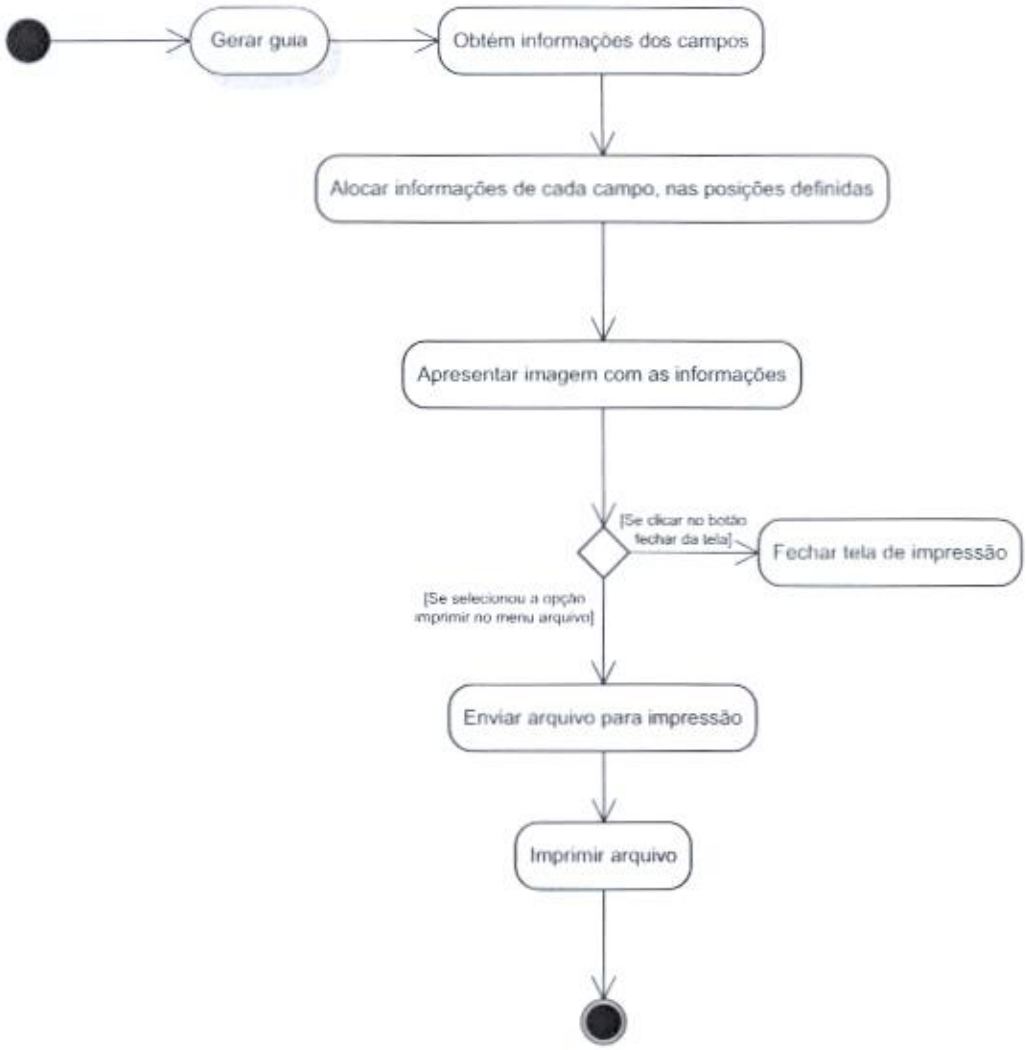
21.3 Diagrama de seqüencia

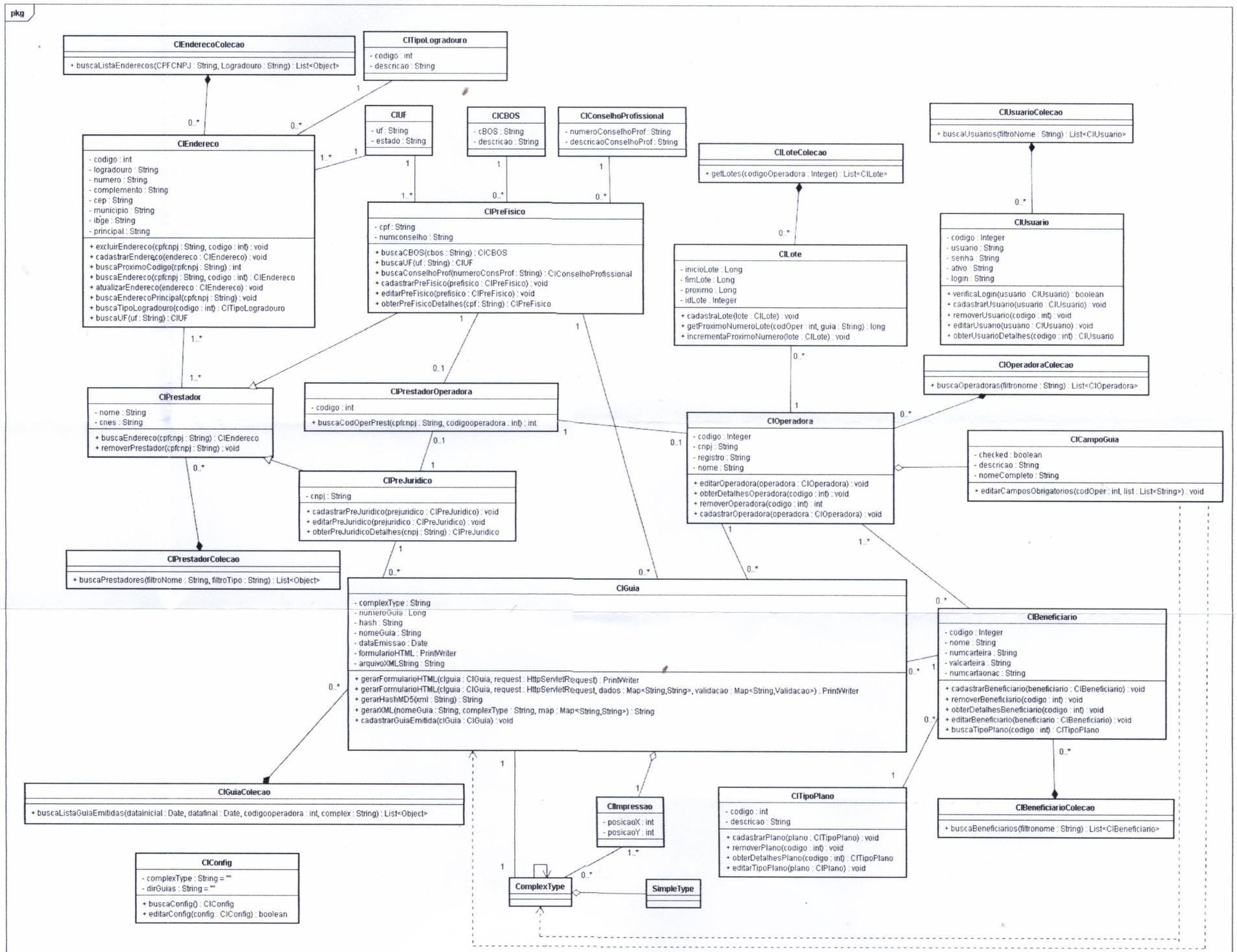


21.4 Diagrama de comunicação

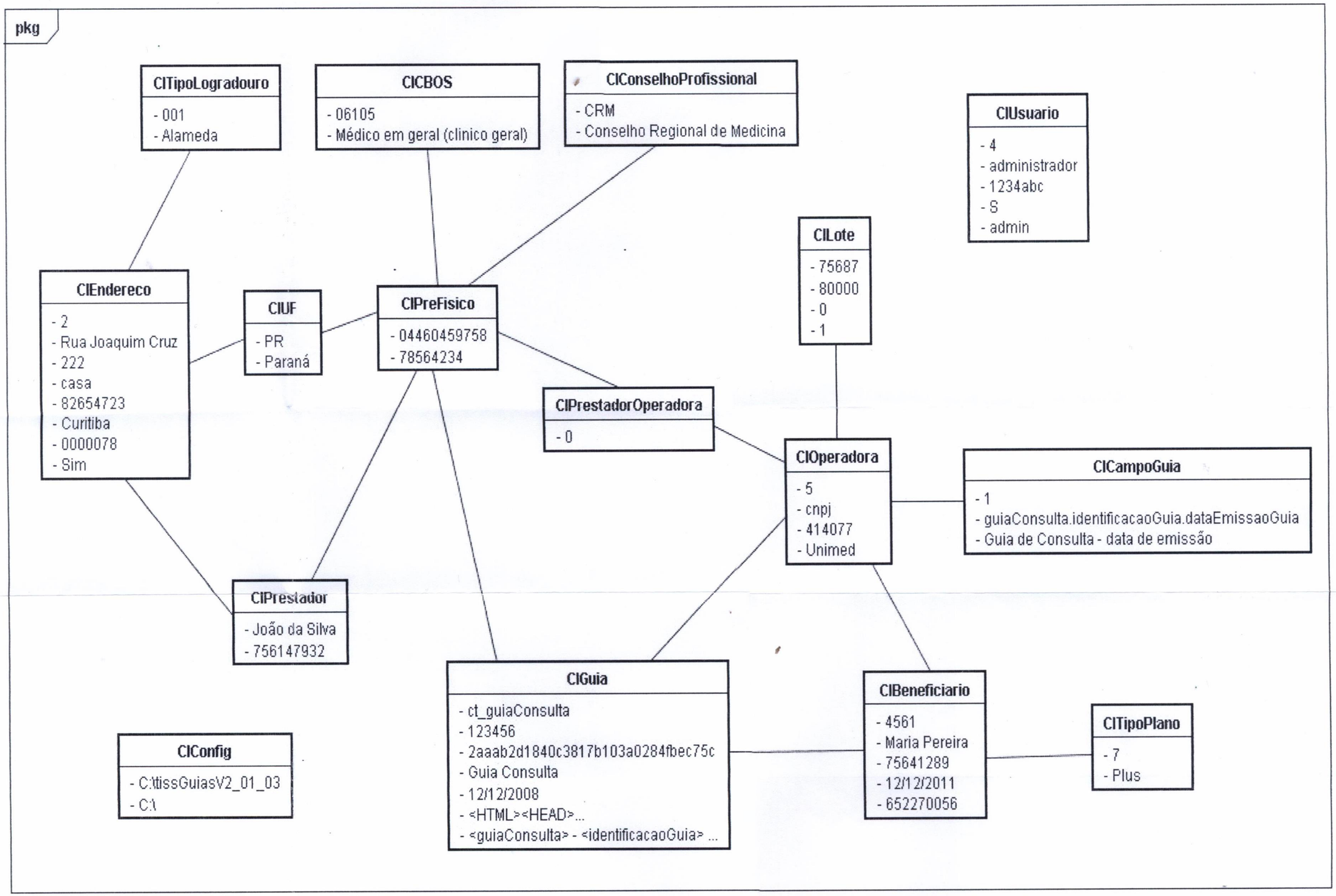


21.5 Diagrama de atividades

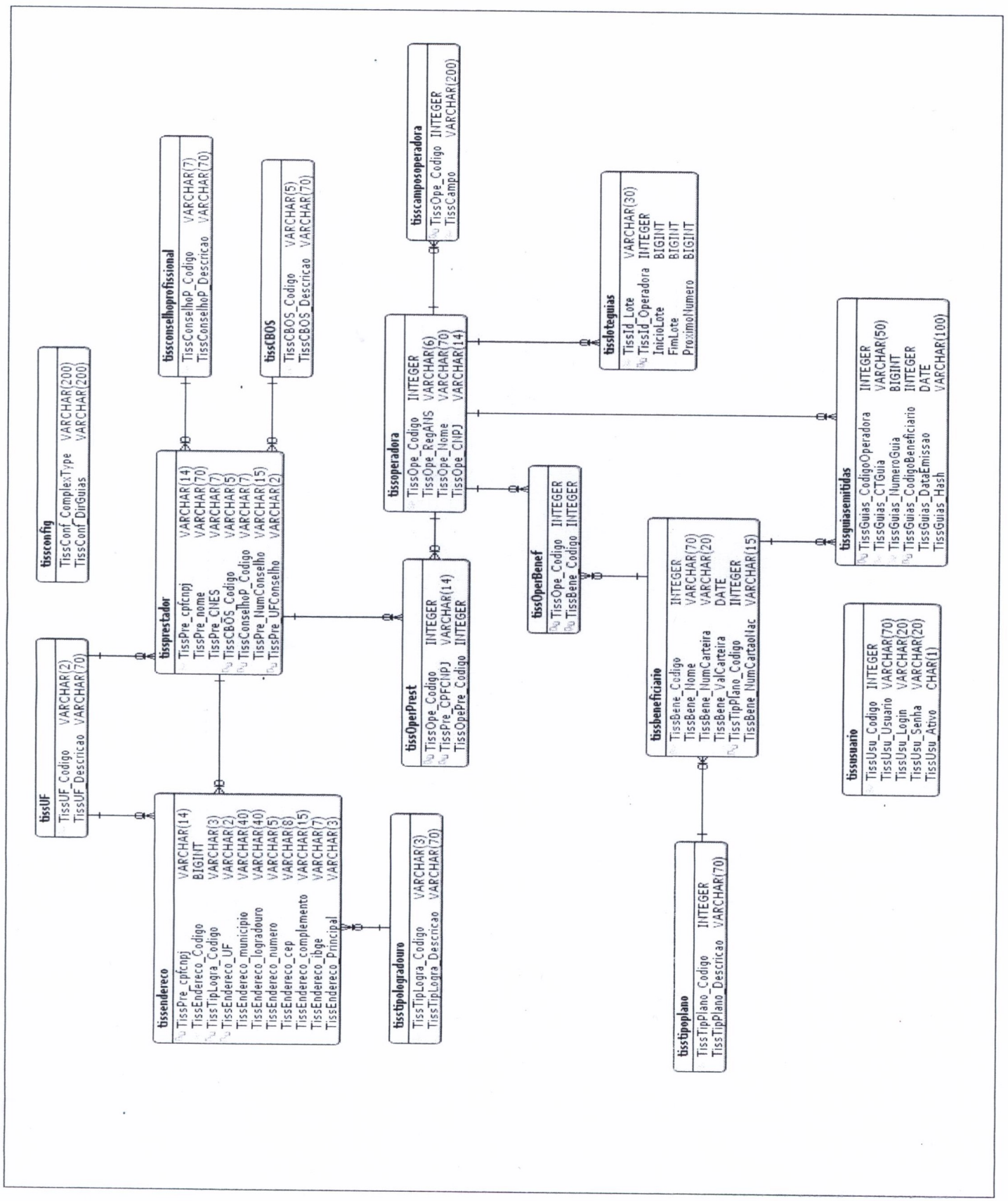



pkg
ClEnderecoColecao
+ buscaListaEnderecos(CPFCNPJ : String, Logradouro : String) : List<Object>
ClTipoLogradouro
- codigo : int
- descricao : String
ClUF
- uf : String
- estado : String
ClCBOS
- cBOS : String
- descricao : String
ClConselhoProfissional
- numeroConselhoProf : String
- descricaoConselhoProf : String
ClEndereco
- codigo : int
- logradouro : String
- numero : String
- complemento : String
- cep : String
- municipio : String
- ibge : String
- principal : String
+ excluirEndereco(cpfcnpj : String, codigo : int) : void
+ cadastrarEndereco(endereco : ClEndereco) : void
+ buscaProximoCodigo(cpfcnpj : String) : int
+ buscaEndereco(cpfcnpj : String, codigo : int) : ClEndereco
+ atualizarEndereco(endereco : ClEndereco) : void
+ buscaEnderecoPrincipal(cpfcnpj : String) : void
+ buscaTipoLogradouro(codigo : int) : ClTipoLogradouro
+ buscaUF(uf : String) : ClUF
ClPreFisico
- cpf : String
- numconselho : String
+ buscaCBOS(cbos : String) : ClCBOS
+ buscaUF(uf : String) : ClUF
+ buscaConselhoProf(numeroConsProf : String) : ClConselhoProfissional
+ cadastrarPreFisico(prefisico : ClPreFisico) : void
+ editarPreFisico(prefisico : ClPreFisico) : void
+ obterPreFisicoDetalhes(cpf : String) : ClPreFisico
ClLoteColecao
+ getLotes(codigoOperadora : Integer) : List<ClLote>
ClLote
- inicioLote : Long
- fimLote : Long
- proximo : Long
- idLote : Integer
+ cadastraLote(lote : ClLote) : void
+ getProximoNumeroLote(codOper : int, guia : String) : long
+ incrementaProximoNumero(lote : ClLote) : void
ClUsuarioColecao
+ buscaUsuarios(filtroNome : String) : List<ClUsuario>
ClUsuario
- codigo : Integer
- usuario : String
- senha : String
- ativo : String
- login : String
+ verificaLogin(usuario : ClUsuario) : boolean
+ cadastrarUsuario(usuario : ClUsuario) : void
+ removerUsuario(codigo : int) : void
+ editarUsuario(usuario : ClUsuario) : void
+ obterUsuarioDetalhes(codigo : int) : ClUsuario
ClOperadoraColecao
+ buscaOperadoras(filtronome : String) : List<ClOperadora>
ClPrestador
- nome : String
- cnes : String
+ buscaEndereco(cpfcnpj : String) : ClEndereco
+ removerPrestador(cpfcnpj : String) : void
ClPrestadorOperadora
- codigo : int
+ buscaCodOperPrest(cpfcnpj : String, codigooperadora : int) : int
ClPreJuridico
- cnpj : String
+ cadastrarPreJuridico(prejuridico : ClPreJuridico) : void
+ editarPreJuridico(prejuridico : ClPreJuridico) : void
+ obterPreJuridicoDetalhes(cnpj : String) : ClPreJuridico
ClOperadora
- codigo : Integer
- cnpj : String
- registro : String
- nome : String
+ editarOperadora(operadora : ClOperadora) : void
+ obterDetalhesOperadora(codigo : int) : void
+ removerOperadora(codigo : int) : int
+ cadastrarOperadora(operadora : ClOperadora) : void
ClCampoGuia
- checked : boolean
- descricao : String
- nomeCompleto : String
+ editarCamposObrigatorios(codOper : int, list : List<String>) : void
ClPrestadorColecao
+ buscaPrestadores(filtroNome : String, filtroTipo : String) : List<Object>
ClGuia
- complexType : String
- numeroGuia : Long
- hash : String
- nomeGuia : String
- dataEmissao : Date
- formularioHTML : PrintWriter
- arquivoXMLString : String
+ gerarFormularioHTML(clguia : ClGuia, request : HttpServletRequest) : PrintWriter
+ gerarFormularioHTML(clguia : ClGuia, request : HttpServletRequest, dados : Map<String,String>, validacao : Map<String,Validacao>) : PrintWriter
+ gerarHashMD5(xml : String) : String
+ gerarXML(nomeGuia : String, complexType : String, map : Map<String,String>) : String
+ cadastrarGuiaEmitida(clGuia : ClGuia) : void
ClBeneficiario
- codigo : Integer
- nome : String
- numcarteira : String
- valcarteira : String
- numcartaonac : String
+ cadastrarBeneficiario(beneficiario : ClBeneficiario) : void
+ removerBeneficiario(codigo : int) : void
+ obterDetalhesBeneficiario(codigo : int) : void
+ editarBeneficiario(beneficiario : ClBeneficiario) : void
+ buscaTipoPlano(codigo : int) : ClTipoPlano
ClGuiaColecao
+ buscaListaGuiaEmitidas(datainicial : Date, datafinal : Date, codigooperadora : int, complex : String) : List<Object>
ClImpressao
- posicaoX : int
- posicaoY : int
ClTipoPlano
- codigo : int
- descricao : String
+ cadastrarPlano(plano : ClTipoPlano) : void
+ removerPlano(codigo : int) : void
+ obterDetalhesPlano(codigo : int) : ClTipoPlano
+ editarTipoPlano(plano : ClPlano) : void
ClBeneficiarioColecao
+ buscaBeneficiarios(filtronome : String) : List<ClBeneficiario>
ClConfig
- complexType : String = ""
- dirGuias : String = ""
+ buscaConfig() : ClConfig
+ editarConfig(config : ClConfig) : boolean
ComplexType
SimpleType

23. DIAGRAMA DE OBJETOS



24. DER


tissUF
TissUF_Codigo VARCHAR(2)
TissUF_Descricao VARCHAR(70)
tissconfig
TissConf_ComplexType VARCHAR(200)
TissConf_DirGuias VARCHAR(200)
tissprestador
TissPre_cpfcnpj VARCHAR(14)
TissPre_nome VARCHAR(70)
TissPre_CNES VARCHAR(7)
TissCBOS_Codigo VARCHAR(5)
TissConselhoP_Codigo VARCHAR(7)
TissPre_NumConselho VARCHAR(15)
TissPre_UFConselho VARCHAR(2)
tissconselhoprofissional
TissConselhoP_Codigo VARCHAR(7)
TissConselhoP_Descricao VARCHAR(70)
tissCBOS
TissCBOS_Codigo VARCHAR(5)
TissCBOS_Descricao VARCHAR(70)
tissoperadora
TissOpe_Codigo INTEGER
TissOpe_RegANS VARCHAR(6)
TissOpe_Nome VARCHAR(70)
TissOpe_CNPJ VARCHAR(14)
tisscamposoperadora
TissOpe_Codigo INTEGER
TissCampo VARCHAR(200)
tissloteguias
TissId_Lote VARCHAR(30)
TissId_Operadora INTEGER
InicioLote BIGINT
FimLote BIGINT
ProximoNumero BIGINT
tissOperPrest
TissOpe_Codigo INTEGER
TissPre_CPFCNPJ VARCHAR(14)
TissOpePre_Codigo INTEGER
tissOperBenef
TissOpe_Codigo INTEGER
TissBene_Codigo INTEGER
tissbeneficiario
TissBene_Codigo INTEGER
TissBene_Nome VARCHAR(70)
TissBene_NumCarteira VARCHAR(20)
TissBene_ValCarteira DATE
TissTipPlano_Codigo INTEGER
TissBene_NumCartaoHac VARCHAR(15)
tissguiasemitidas
TissGuias_CodigoOperadora INTEGER
TissGuias_CTGuia VARCHAR(50)
TissGuias_NumeroGuia BIGINT
TissGuias_CodigoBeneficiario INTEGER
TissGuias_DataEmissao DATE
TissGuias_Hash VARCHAR(100)
tissendereco
TissPre_cpfcnpj VARCHAR(14)
TissEndereco_Codigo BIGINT
TissTipLogra_Codigo VARCHAR(3)
TissEndereco_UF VARCHAR(2)
TissEndereco_municipio VARCHAR(40)
TissEndereco_logradouro VARCHAR(40)
TissEndereco_numero VARCHAR(5)
TissEndereco_cep VARCHAR(8)
TissEndereco_complemento VARCHAR(15)
TissEndereco_ibge VARCHAR(7)
TissEndereco_Principal VARCHAR(3)
tisstipologradouro
TissTipLogra_Codigo VARCHAR(3)
TissTipLogra_Descricao VARCHAR(70)
tisstipoplano
TissTipPlano_Codigo INTEGER
TissTipPlano_Descricao VARCHAR(70)
tissusuario
TissUsu_Codigo INTEGER
TissUsu_Usuario VARCHAR(70)
TissUsu_Login VARCHAR(20)
TissUsu_Senha VARCHAR(20)
TissUsu_Ativo CHAR(1)

25. CÓDIGO FONTE

Código Fonte

Telas e Scripts

Nome do arquivo: aba.css

```
div.jAbaContainer { padding:3px; }
div.jAbaHeader { font-size: 13px;}
div.jAbaBody {border: solid 1px #CCCCCC; height:100%; overflow-y:scroll;}
div.jAbaHeader .jTabHeaderHidden {
        cursor: pointer;
        background-image: url(./images/abamiddlenormal.gif);
        background-repeat: repeat-x;
        border-left: solid 1px #CCCCCC;
        border-right: solid 1px #CCCCCC;
        padding: 3px;
}
div.jAbaHeader .jTabHeaderInline {
        cursor: pointer;
        background-image: url(./images/abamiddleclick.gif);
        background-repeat: repeat-x;
        border-left: solid 1px #CCCCCC;
        border-right: solid 1px #CCCCCC;
        padding: 3px;
}
div.jAbaBody .jTabBodyHidden { display: none;}
div.jAbaBody .jTabBodyInline{
        display: inline;
        width:100px;
        height:100px;
}
```

Nome do arquivo: aba.html

```
<script src="aba.js" type="text/javascript"></script>
<link href="aba.css" type="text/css" rel="stylesheet">
<div id="divFora">
<div id="divDentro1"> div dentro 1 <input type="text" size="20"> </div>
<div id="divDentro2"> div dentro 2 </div>
<div id="divDentro3"> div dentro 3</div>
<div id="divDentro4"> div dentro 4</div>
</div>
<script>
        var aba = new JTabControl("divFora",500,300);
        aba.addTab("divDentro1", "Título 1");
        aba.addTab("divDentro2", "Título 2");
        aba.addTab("divDentro3", "Título 3");
        aba.addTab("divDentro4", "Título 4");
</script>
```

Nome do arquivo: cabecalho.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
        <CENTER><img src="img/write.png" align=middle ></CENTER>
</body>
</html>
```

Nome do arquivo: cadastroBen.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<!-- Javascript do Calendario -->
<script language="JavaScript" src="calendar1.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
<title>Cadastro de Beneficiários</title>
</head>
<body>
<form id="formBeneficiario" name="formBeneficiario" method="post"
action="SvtBeneficiarioCadastro.ts" class="niceform">
<input type="hidden" name="operacao" value="${operacao}"/>
<input type="hidden" name="codigo" value="${beneficiario.codigo}"/>

<h3> Beneficiário </h3>
<fieldset><legend>Beneficiario</legend>
<div class="linha1"><label>Nome: <input name="nome"
        type="text" id="nome" size="70" maxlength="70" value="${beneficiario.nome}">
</label>
</div>
<div class="linha2"><label>Número da carteira: <input
        type="text" name="numcarteira" id="numcarteira" size="20" maxlength="20"
        value="${beneficiario.numcarteira}"> </label>
        <label>Validade da carteira: <input name="valcarteira" type="text" id="valcarteira"
        size="8" readonly="readonly" value="${beneficiario.valcarteira}">
        <a href="javascript:cal1.popup();"><img src="img/cal.gif" width="16" height="16"
border="0" alt="Clique para selecionar a data"></a>
        <br>
        </label>
</div>

<div class="linha3"><label>Tipo do plano: <input
        name="tpplano" type="text" id="tpplano" size="40" maxlength="40"
value="${beneficiario.tpplano}">
        </label>
</div>
<div>
        <label>Número do Cartão Nacional: <input name="numcartaonac"
        type="text" id="numcartaonac" size="15" maxlength="15"
value="${beneficiario.numcartaonac}">
```

```
        </label>
</div>

<div class="linhabotoes" align="left"><input
        name="SalvarCadastro" type="submit" value="Salvar">
        <input type="Button" name="cancelar"
        value="Cancelar" onclick="history.go(-1)">
</div>
</fieldset>

</form>
<script>
        var cal1 = new calendar1(
        document.forms['formBeneficiario'].elements['valcarteira']);
</script>
</body>
</html>
```

Nome do arquivo: cadastroEndereco.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Endereço</title>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
<form name="formendereco" id="formendereco" method="post" action="" class="niceform">
<input type="hidden" name="operacao" value="${operacao}"/>
<input name="cpfcnpj" type="hidden" id="cpfcnpj" value="${cpfcnpj}">
<input name="codigo" type="hidden" id="codigo" value="${codigo}">

<h3> Cadastro de Endereços </h3>
<fieldset><legend>Endereço</legend>
        <div class="endereco01">
<label>Tipo logradouro
<select name="tipologradouro">
<option value="">Selecione</option>
<c:forEach var="tipologradouroV" items="${listaTipoLogradouro}">
        <c:if test="${objendereco.tipoLogradouro==tipologradouroV}">
                <option value="${tipologradouroV}" selected>${tipologradouroV}</option>
        </c:if>
        <c:if test="${objendereco.tipoLogradouro!=tipologradouroV}">
                <option value="${tipologradouroV}">${tipologradouroV}</option>
        </c:if>
</c:forEach>
</select>
</label>
<label>Logradouro
   <input name="logradouro" type="text" id="logradouro" size="40"
        maxlength="40" value="${objendereco.logradouro}"> </label>
        </div>
<label>Endereço Principal
        <select name="principal">
```

```
                <option value="SIM">Sim</option>
                <option value="NAO">Não</option>
        </select>
</label>
<div class="endereco02"><label>Número <input name="numero"
        type="text" id="numero" size="05" maxlength="05"
        value="${objendereco.numero}"> </label> <label>Complemento <input
        name="complemento" type="text" id="complemento" size="15"
        maxlength="15" value="${objendereco.complemento}"> </label> <label>CEP
<input name="cep" type="text" id="cep" size="08" maxlength="08"
        value="${objendereco.cep}"> </label></div>

<div class="endereco03"><label>Municipio: <input
        name="municipio" type="text" id="municipio" size="40" maxlength="40"
        value="${objendereco.municipio}"> </label> <label>U.F. <input
        name="uf" type="text" id="uf" size="02" maxlength="02"
        value="${objendereco.uf }"> </label> <label>IBGE <input
        name="ibge" type="text" id="ibge" size="07" maxlength="07"
        value="${objendereco.ibge}"> </label></div>
<div class="linhabotoes">
        <input name="SalvarCadastro" type="submit" value="Salvar">
   <input type="Button" name="cancelar" value="Cancelar" onclick="history.go(-1)">
</div>
</fieldset>
</form>
</body>
</html>
```

Nome do arquivo: cadastroOpe.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<script src="js/aba.js" type="text/javascript"></script>
<script language="JavaScript" src="js/funcoesPrestador.js"></script>
<script language="JavaScript" src="js/funcoesOperadora.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
<link href="aba.css" type="text/css" rel="stylesheet">
<title>Cadastro :: Operadoras</title>
</head>
<body>
<form id="form1" name="cadastro" method="post"        action="SvtOperadoraCadastro.ts"
onsubmit="return valida_cnpj(this.cnpj.value)" class="niceform">
<input type="hidden" name="operacao" value="${operacao }"/>
<input type="hidden" name="codigo" value="${operadora.codigo }"/>
        <h3> Cadastro de Operadoras </h3>
        <fieldset><legend>Operadora</legend>
                <div class="linha1"><label>Número do Registro ANS: <input
                        name="registro" type="text" id="registro" size="06" maxlength="06"
value="${operadora.registro}"></label>
                        <label>CNPJ: <input name="cnpj" type="text" id="cnpj" size="14"
maxlength="14" value="${operadora.cnpj}"></label>
                </div>
                <div><label>Nome: <input name="nome" type="text" id="nome"
```

```
                size="70" maxlength="70" value="${operadora.nome }"> </label>
            </div>
            <br>
            <br>
            <br>
            <fieldset><legend>Parâmetros da Operadora</legend>
                <div>
                    <label>Identificação do Prestador</label>
                    <select name="idprestador">
                        <c:if test="${operadora.idPrestador=='0'}">
                        <option value="0" selected>CPF/CNPJ</option>
                        <option value="1">Código na Operadora</option>
                        </c:if>
                        <c:if test="${operadora.idPrestador=='1'}">
                        <option value="0">CPF/CNPJ</option>
                        <option value="1" selected>Código na
Operadora</option>
                        </c:if>
                    </select>
                </div>
                <br>
                <div id="divFora">
                <div id="divGuiaConsulta">
                    <table>
                    <c:forEach var="campo" items="${camposGuiaConsulta}">
                        <tr>
                        <c:if test="${campo.checked}">
                            <td>${campo.nomeCompleto}</td><td><input
id="guiaConsulta" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}" checked></td>
                        </c:if>
                        <c:if test="${!campo.checked}">
                            <td>${campo.nomeCompleto}</td><td><input
id="guiaConsulta" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}"></td>
                        </c:if>
                        </tr>
                    </c:forEach>
                    </table>
                    <a href="javascript:selecionar_tudo('guiaConsulta')">Marcar
todos</a>
                    <a
href="javascript:deselecionar_tudo('guiaConsulta')">Desmarcar Todos</a>

                </div>
                <div id="divGuiaSP_SADT" >
                    <table>
                    <c:forEach var="campo" items="${camposGuiaSP_SADT}">
                        <tr>
                        <c:if test="${campo.checked}">
                            <td>${campo.nomeCompleto}</td><td><input
id="guiaSP_SADT" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}" checked></td>
                        </c:if>
                        <c:if test="${!campo.checked}">
                            <td>${campo.nomeCompleto}</td><td><input
id="guiaSP_SADT" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}"></td>
                        </c:if>
                                </tr>
```

```
                                    </c:forEach>
                                    </table>
                                    <a href="javascript:selecionar_tudo('guiaSP_SADT')">Marcar
todos</a>
                                    <a
href="javascript:deselecionar_tudo('guiaSP_SADT')">Desmarcar Todos</a>
                            </div>
                            <div id="divSolicInternacao">
                                    <table>
                                    <c:forEach var="campo"
items="${camposGuiaSolicInternacao}">
                                            <tr>
                                            <c:if test="${campo.checked}">
                                                    <td>${campo.nomeCompleto}</td><td><input
id="guiaSolicInternacao" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}" checked></td>
                                            </c:if>
                                            <c:if test="${!campo.checked}">
                                                    <td>${campo.nomeCompleto}</td><td><input
id="guiaSolicInternacao" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}"></td>
                                            </c:if>
                                                            </tr>
                                    </c:forEach>
                                    </table>
                                    <a
href="javascript:selecionar_tudo('guiaSolicInternacao')">Marcar todos</a>
                                    <a
href="javascript:deselecionar_tudo('guiaSolicInternacao')">Desmarcar Todos</a>
                            </div>
                            <div id="divResumoInternacao">
                                    <table>
                                    <c:forEach var="campo"
items="${camposGuiaResumoInternacao}">
                                            <tr>
                                            <c:if test="${campo.checked}">
                                                    <td>${campo.nomeCompleto}</td><td><input
id="guiaResumoInternacao" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}" checked></td>
                                            </c:if>
                                            <c:if test="${!campo.checked}">
                                                    <td>${campo.nomeCompleto}</td><td><input
id="guiaResumoInternacao" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}"></td>
                                            </c:if>
                                                            </tr>
                                    </c:forEach>
                                    </table>
                                    <a
href="javascript:selecionar_tudo('guiaResumoInternacao')">Marcar todos</a>
                                    <a
href="javascript:deselecionar_tudo('guiaResumoInternacao')">Desmarcar Todos</a>
                            </div>
                            <div id="divHonorarioIndividual">
                                    <table>
                                    <c:forEach var="campo"
items="${camposGuiaHonorarioIndividual}">
                                            <tr>
                                            <c:if test="${campo.checked}">
```

```
                                        <td>${campo.nomeCompleto}</td><td><input
id="guiaHonorarioIndividual" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}" checked></td>
                                    </c:if>
                                    <c:if test="${!campo.checked}">
                                        <td>${campo.nomeCompleto}</td><td><input
id="guiaHonorarioIndividual" type="checkbox" name="obr.${campo.nomeCompleto}"
title="${campo.descricao}"></td>
                                    </c:if>
                                        </tr>
                                </c:forEach>
                                </table>
                                <a
href="javascript:selecionar_tudo('guiaHonorarioIndividual')">Marcar todos</a>
                                <a
href="javascript:deselecionar_tudo('guiaHonorarioIndividual')">Desmarcar Todos</a>
                            </div>
                            </div>
                            <div id="divLotes">
                                <label>Lote de
guias</label>    <label>Inicio:<input name="iniciolote" type="text"
maxlength="12" size="12"></label>

                                        <label>Fim:<input name="fimlote" type="text"
maxlength="12" size="12"></label>
                                <c:if test="${!empty lotes}">
                                <table border="1">
                                    <tr>
                                        <th>Inicio do Lote</th><th>Fim do
Lote</th><th>Próximo número disponivel</th>
                                    </tr>
                                        <c:forEach var="lote" items="${lotes}">
                                            <tr>

        <td>${lote.inicioLote}</td><td>${lote.fimLote}</td><td>${lote.proximo}</td>
                                            </tr>
                                        </c:forEach>
                                </table>
                                </c:if>
                            </div>
                            <script>
                                var aba = new JTabControl("divFora",600,700);

                                aba.addTab("divGuiaConsulta", "Guia de Consulta");
                                aba.addTab("divGuiaSP_SADT", "Guia SP/SADT");
                                aba.addTab("divSolicInternacao", "Guia Solicitação de
Internação");
                                aba.addTab("divResumoInternacao", "Guia Resumo de
Internação");
                                aba.addTab("divHonorarioIndividual", "Guia Honorário
Individual");
                                aba.addTab("divLotes", "Lotes de guias");
                            </script>
    </fieldset>
                <div class="linhabotoes">
                    <input  type="submit" name="SalvarCadastro" value="Salvar">
                    <input type="Button" name="cancelar" value="Cancelar"
onclick="history.go(-1)">
                </div>
        </fieldset>
```

```
</form>
</body>
</html>
```

Nome do arquivo: cadastroPreFisico.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="js/validacao.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Cadastro :: Prestador Fisico</title>
</head>
<body>
<form name="formFisico" id="formFisico" method="post"
action="SvtPrestadorFisicoCadastro.ts" class="niceform">
<input type="hidden" name="operacao" value="${operacao}"/>
<input type="hidden" name="tipo" value="fisico"/>
<h3> Cadastro de Prestadores </h3>
<fieldset>
<legend>Prestador Fisico</legend>
<div class="linha01">
<label>Nome
<input name="nome" type="text" id="nome" size="70" maxlength="70"
value="${prefisico.nome}">
</label>
</div>
<div class="linha02">
<label>CPF
<input name="cpf" type="text" id="cpf" size="11" maxlength="11" value="${prefisico.cpf}">
</label>
<label>CNES
<input name="cnes" type="text" id="cnes" size="7" maxlength="07" value="${prefisico.cnes}">
</label>
<label>Código Operadora
<input name="codoperadora" type="text" id="codoperadora" size="14" maxlength="14"
value="${prefisico.codoperadora}">
</label>
<label>CBO-s
<input name="cbos" type="text" id="cbos" size="05" maxlength="05" value="${prefisico.cbos}">
</label>
</div>
<div class="linha03">
<label>Conselho Profissional
<select name="conselhoprof" >
<option value="" selected> </option>
<c:forEach var="conselho" items="${listaConselhos}">
        <c:if test="${prefisico.conselhoprof==conselho}">
                <option value="${conselho}" selected>${conselho}</option>
        </c:if>
        <c:if test="${prefisico.conselhoprof!=conselho}">
                <option value="${conselho}">${conselho}</option>
        </c:if>
</c:forEach>
```

```
</select>
</label>
<label>N° no conselho:
<input name="numconselho" type="text" id="numconselho" size="15" maxlength="15"
value="${prefisico.numconselho}">
</label>
<label>U.F. do Conselho Profissional
<input name="ufconselho" type="text" id="ufconselho" size="02" maxlength="02"
value="${prefisico.ufconselho}">
</label>
</div>
<P>
<FIELDSET>
<LEGEND>Endereço Principal</LEGEND>
<div class="endereco01">
<label>Tipo logradouro
<select name="tipologradouro">
<option value="">Selecione</option>
<c:forEach var="tipologradouroV" items="${listaTipoLogradouro}">
        <c:if test="${prefisico.endereco.tipoLogradouro==tipologradouroV}">
                <option value="${tipologradouroV}" selected>${tipologradouroV}</option>
        </c:if>
        <c:if test="${prefisico.endereco.tipoLogradouro!=tipologradouroV}">
                <option value="${tipologradouroV}">${tipologradouroV}</option>
        </c:if>
</c:forEach>
</select>
</label>
<label>Logradouro
<input name="logradouro" type="text" id="logradouro" size="40" maxlength="40"
value="${prefisico.endereco.logradouro}">
</label>
</div>
<div class="endereco02">
<label>Número
<input name="numero" type="text" id="numero" size="05" maxlength="05"
value="${prefisico.endereco.numero}">
</label>
<label>Complemento
<input name="complemento" type="text" id="complemento" size="15" maxlength="15"
value="${prefisico.endereco.complemento}">
</label>
<label>CEP
<input name="cep" type="text" id="cep" size="08" maxlength="08"
value="${prefisico.endereco.cep}">
</label>
</div>
<div class="endereco03">
<label>Município:
<input name="municipio" type="text" id="municipio" size="40" maxlength="40"
value="${prefisico.endereco.municipio}">
</label>
<label>U.F.
<input name="uf" type="text" id="uf" size="02" maxlength="02" value="${prefisico.endereco.uf
}">
</label>
<label>IBGE
<input name="ibge" type="text" id="ibge" size="07" maxlength="07"
value="${prefisico.endereco.ibge}">
</label>
```

```
<input name="CadastrarEndereco" type="submit" value="Endereco">
</div>
</FIELDSET>
<div class="linhabotoes" align="center">
        <input name="SalvarCadastro" type="submit" value="Salvar">
        <input type="Button" name="cancelar" value="Cancelar" onclick="history.go(-1)">
</div>
</fieldset>
</form>
</body>
</html>
```

Nome do arquivo: cadastroPreJuridico.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
<form name="formjuridico" id="formjuridico" method="post" action="" class="niceform">
<input type="hidden" name="operacao" value="${operacao}"/>
<input type="hidden" name="tipo" value="juridico"/>
<h3> Cadastro de Prestadores </h3>
<fieldset><legend>Prestador Jurídico</legend>
        <div class="linha01"><label>Nome <input name="nome"
        id="nome" type="text" size="70" maxlength="70" value="${prejuridico.nome}"> </label>
        </div>
        <div class="linha02"><label>CNPJ <input name="cnpj"
        id="cnpj" type="text" size="14" maxlength="14" value="${prejuridico.cnpj}"> </label>
        </div>
        <div class="linha03"><label>CNES <input name="cnes"
        id="cnes" type="text" size="07" maxlength="07" value="${prejuridico.cnes}"> </label>
        </div>
        <div class="linha04"><label>Código na operadora <input
        name="codoperadora" id="codoperadora" type="text" size="14"
        maxlength="14" value="${prejuridico.codoperadora}"> </label>
        </div>
<P>
<FIELDSET>
<LEGEND>Endereço Principal</LEGEND>
        <div class="endereco01"><label>Tipo logradouro
<select name="tipologradouro">
<option value="">Selecione</option>
<c:forEach var="tipologradouroV" items="${listaTipoLogradouro}">
        <c:if test="${prejuridico.endereco.tipoLogradouro==tipologradouroV}">
                <option value="${tipologradouroV}" selected>${tipologradouroV}</option>
        </c:if>
        <c:if test="${prejuridico.endereco.tipoLogradouro!=tipologradouroV}">
                <option value="${tipologradouroV}">${tipologradouroV}</option>
        </c:if>
</c:forEach>
</select>
```

```
</label>
<label>Logradouro
   <input name="logradouro" type="text" id="logradouro" size="40"
        maxlength="40" value="${prejuridico.endereco.logradouro}"> </label>
        </div>
<div class="endereco02"><label>Número <input name="numero"
        type="text" id="numero" size="05" maxlength="05"
        value="${prejuridico.endereco.numero}"> </label> <label>Complemento <input
        name="complemento" type="text" id="complemento" size="15"
        maxlength="15" value="${prejuridico.endereco.complemento}"> </label> <label>CEP
<input name="cep" type="text" id="cep" size="08" maxlength="08"
        value="${prejuridico.endereco.cep}"> </label></div>
<div class="endereco03"><label>Município: <input
        name="municipio" type="text" id="municipio" size="40" maxlength="40"
        value="${prejuridico.endereco.municipio}"> </label> <label>U.F. <input
        name="uf" type="text" id="uf" size="02" maxlength="02"
        value="${prejuridico.endereco.uf }"> </label> <label>IBGE <input
        name="ibge" type="text" id="ibge" size="07" maxlength="07"
        value="${prejuridico.endereco.ibge}"> </label>
<input name="CadastrarEndereco" type="submit" value="Endereco">
</div>
</FIELDSET>
<div class="linhabotoes">
        <input name="SalvarCadastro" type="submit" value="Salvar">
   <input type="Button" name="cancelar" value="Cancelar" onclick="history.go(-1)">
</div>
</fieldset>
</form>
</body>
</html>
```

Nome do arquivo: cadastroUsu.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Cadastro :: Usuários</title>
<script language="JavaScript" src="js/funcoesUsuario.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
<form id="form1" name="form1" method="post" action="SvtUsuarioCadastro.ts">
<h3> Cadastro de Usuários </h3>
<fieldset><legend>Usuário</legend>
<label>Nome: </label>
<input name="nome"    type="text" id="nome" size="70" maxlength="70"
value="${usuario.usuario}">
<label>Código:</label>
 <input name="codigo" type="text" id="codigo" size="10" maxlength="10"
        readonly="readonly" value="${usuario.codigo}">
<div class="linha2"><label>Login: <input name="login"
        type="text" id="login" size="40" maxlength="20" value="${usuario.login}"> </label>
</div>
<div class="linha3"><label>Senha: <input name="senha"
```

```
        type="password" id="senha" size="12" maxlength="20"> </label>
        <label>Confirme a senha:
<input name="confsenha" type="password" id="confsenha"      size="12"> </label>
</div>
<div class="linha"><label>Situação: <input name="at"
        type="checkbox" value="Ativo"
        <c:if test="${usuario.ativo=='S'}">checked</c:if> />Ativo</label></div>
<div class="linhabotoes">
        <input type="Submit" name="salvar"
        value="Salvar" onclick="return verificaSenhas()">
        <input type="Button" name="cancelar"
        value="Cancelar" onclick="history.go(-1)">
</div>
</fieldset>
</form>
</body>
</html>
```

Nome do arquivo: calendar.html

```
<!--
Title: Tigra Calendar
URL: http://www.softcomplex.com/products/tigra_calendar/
Version: 3.4 (all formats)
Date:  01/06/2007
Note: Permission given to use this script in ANY kind of applications if
  header lines are left unchanged.
Note: Script consists of two files: calendar?.js and calendar.html
-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Selecione a data desejada!</title>
<style>
        td {font-family: Tahoma, Verdana, sans-serif; font-size: 12px;}
        td a {text-decoration: none;}
</style>
<script language="JavaScript">

// months as they appear in the calendar's title
var ARR_MONTHS = ["Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",
                  "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro"];
// week day titles as they appear on the calendar
var ARR_WEEKDAYS = ["D", "S", "T", "Q", "Q", "S", "S"];
// day week starts from (normally 0-Su or 1-Mo)
var NUM_WEEKSTART = 1;
// path to the directory where calendar images are stored. trailing slash req.
var STR_ICONPATH = 'img/';

var re_urlCur = new RegExp('c=(\\-?\\d+)');
var re_urlSel = new RegExp('s=(\\-?\\d+)');
var re_id = new RegExp('id=(\\d+)');

var dt_selected = (re_urlSel.exec(String(window.location))
        ? new Date(new Number(RegExp.$1)) : new Date());
var dt_current = (re_urlCur.exec(String(window.location))
        ? new Date(new Number(RegExp.$1)) : dt_selected);
```

```
var num_id = (re_id.exec(String(window.location))
        ? new Number(RegExp.$1) : 0);

var obj_caller = (window.opener ? window.opener.calendars[num_id] : null);

if (obj_caller && obj_caller.year_scroll) {
        // get same date in the previous year
        var dt_prev_year = new Date(dt_current);
        dt_prev_year.setFullYear(dt_prev_year.getFullYear() - 1);
        if (dt_prev_year.getDate() != dt_current.getDate())
                dt_prev_year.setDate(0);

        // get same date in the next year
        var dt_next_year = new Date(dt_current);
        dt_next_year.setFullYear(dt_next_year.getFullYear() + 1);
        if (dt_next_year.getDate() != dt_current.getDate())
                dt_next_year.setDate(0);
}

// get same date in the previous month
var dt_prev_month = new Date(dt_current);
if (dt_prev_month.getMonth()) {
        dt_prev_month.setMonth(dt_prev_month.getMonth() - 1);
        if (dt_prev_month.getDate() != dt_current.getDate())
                dt_prev_month.setDate(0);
}
else {
        dt_prev_month.setFullYear(dt_prev_month.getFullYear() - 1);
        dt_prev_month.setMonth(11);
}

// get same date in the next month
var dt_next_month = new Date(dt_current);
dt_next_month.setMonth(dt_next_month.getMonth() + 1);
if (dt_next_month.getDate() != dt_current.getDate())
        dt_next_month.setDate(0);

// get first day to display in the grid for current month
var dt_firstday = new Date(dt_current);
dt_firstday.setDate(1);
dt_firstday.setDate(1 - (7 + dt_firstday.getDay() - NUM_WEEKSTART) % 7);

// function passing selected date to calling window
function set_datetime(n_datetime, b_close) {
        if (!obj_caller) return;

        var dt_datetime = obj_caller.prs_time(
                (document.cal ? document.cal.time.value : ''),
                new Date(n_datetime)
        );

        if (!dt_datetime) return;
        if (b_close) {

                obj_caller.target.value = (document.cal
                        ? obj_caller.gen_tsmp(dt_datetime)
                        : obj_caller.gen_date(dt_datetime)
                );window.close();
        }
        else obj_caller.popup(dt_datetime.valueOf());
```

```
}

</script>
</head>
<body bgcolor="#FFFFFF" marginheight="5" marginwidth="5" topmargin="5" leftmargin="5"
rightmargin="5">
<table class="clsOTable" cellspacing="0" border="0" width="100%">
<tr><td bgcolor="#4682B4">
<table cellspacing="1" cellpadding="3" border="0" width="100%">
<tr><td colspan="7"><table cellspacing="0" cellpadding="0" border="0" width="100%">
<tr>
<script language="JavaScript">
document.write(
'<td nowrap>'+(obj_caller&&obj_caller.year_scroll?'<a
href="javascript:set_datetime('+dt_prev_year.valueOf()+')"><img
src="'+STR_ICONPATH+'prev_year.gif" width="16" height="16" border="0" alt="Ano
anterior"></a> ":")+'<a href="javascript:set_datetime('+dt_prev_month.valueOf()+')"><img
src="'+STR_ICONPATH+'prev.gif" width="16" height="16" border="0" alt="Mês
Anterior"></a></td>'+
'<td align="center" width="100%"><font
color="#ffffff">'+ARR_MONTHS[dt_current.getMonth()]+' '+dt_current.getFullYear() +
'</font></td>'+
'<td nowrap><a href="javascript:set_datetime('+dt_next_month.valueOf()+')"><img
src="'+STR_ICONPATH+'next.gif" width="16" height="16" border="0" alt="Próximo
mês"></a>'+(obj_caller && obj_caller.year_scroll?' <a
href="javascript:set_datetime('+dt_next_year.valueOf()+')"><img
src="'+STR_ICONPATH+'next_year.gif" width="16" height="16" border="0" alt="Próximo
Ano"></a>":")+'</td>'
);
</script>
</tr>
</table></td></tr>
<tr>
<script language="JavaScript">

// print weekdays titles
for (var n=0; n<7; n++)
        document.write('<td bgcolor="#87cefa" align="center"><font
color="#ffffff">'+ARR_WEEKDAYS[(NUM_WEEKSTART+n)%7]+'</font></td>');
document.write('</tr>');

// print calendar table
var dt_current_day = new Date(dt_firstday);
while (dt_current_day.getMonth() == dt_current.getMonth() ||
        dt_current_day.getMonth() == dt_firstday.getMonth()) {
        // print row heder
        document.write('<tr>');
        for (var n_current_wday=0; n_current_wday<7; n_current_wday++) {
                if (dt_current_day.getDate() == dt_selected.getDate() &&
                        dt_current_day.getMonth() == dt_selected.getMonth() &&
                        dt_current_day.getFullYear() == dt_selected.getFullYear())
                        // print current date
                        document.write('<td bgcolor="#ffb6c1" align="center" width="14%">');
                else if (dt_current_day.getDay() == 0 || dt_current_day.getDay() == 6)
                        // weekend days
                        document.write('<td bgcolor="#dbeaf5" align="center" width="14%">');
                else
                        // print working days of current month
                        document.write('<td bgcolor="#ffffff" align="center" width="14%">');
```

```
                document.write('<a href="javascript:set_datetime('+dt_current_day.valueOf()
+', true);">');

                if (dt_current_day.getMonth() == this.dt_current.getMonth())
                        // print days of current month
                        document.write('<font color="#000000">');
                else
                        // print days of other months
                        document.write('<font color="#606060">');

                document.write(dt_current_day.getDate()+'</font></a></td>');
                dt_current_day.setDate(dt_current_day.getDate()+1);
        }
        // print row footer
        document.write('</tr>');
}
if (obj_caller && obj_caller.time_comp) {
        document.write('<form onsubmit="javascript:set_datetime('+dt_current.valueOf()+',
true)" name="cal"><tr><td colspan="7" bgcolor="#87CEFA"><font color="White" face="tahoma,
verdana" size="2">Time: <input type="text" name="time"
value="'+obj_caller.gen_time(this.dt_current)+'" size="8"
maxlength="8"></font></td></tr></form>');
        document.forms['cal'].elements['time'].focus();
}
</script>
</table>
</td></tr></table>
</body>
</html>
```

Nome do arquivo: calendar1.js

```
// Title: Tigra Calendar
// URL: http://www.softcomplex.com/products/tigra_calendar/
// Version: 3.4 (European date format)
// Date: 07/12/2007
// Note: Permission given to use this script in ANY kind of applications if
//    header lines are left unchanged.
// Note: Script consists of two files: calendar?.js and calendar.html

// if two digit year input dates after this year considered 20 century.
var NUM_CENTYEAR = 30;
// is time input control required by default
var BUL_TIMECOMPONENT = false;
// are year scrolling buttons required by default
var BUL_YEARSCROLL = true;

var calendars = [];
var RE_NUM = /^\-?\d+$/;

function calendar1(obj_target) {

        // assigning methods
        this.gen_date = cal_gen_date1;
        this.gen_time = cal_gen_time1;
        this.gen_tsmp = cal_gen_tsmp1;
        this.prs_date = cal_prs_date1;
        this.prs_time = cal_prs_time1;
        this.prs_tsmp = cal_prs_tsmp1;
```

```
        this.popup    = cal_popup1;

        // validate input parameters
        if (!obj_target)
                return cal_error("Error calling the calendar: no target control specified");
        if (obj_target.value == null)
                return cal_error("Error calling the calendar: parameter specified is not valid
target control");
        this.target = obj_target;
        this.time_comp = BUL_TIMECOMPONENT;
        this.year_scroll = BUL_YEARSCROLL;

        // register in global collections
        this.id = calendars.length;
        calendars[this.id] = this;
}

function cal_popup1 (str_datetime) {
        if (str_datetime)
                this.dt_current = this.prs_tsmp(str_datetime);
        else
                this.dt_selected = this.dt_current = this.prs_tsmp(this.target.value);

        if (!this.dt_current) return;

        var obj_calwindow = window.open(
                'calendar.html?id=' + this.id + '&s=' + this.dt_selected.valueOf() + '&c=' +
this.dt_current.valueOf(),
                'Calendar', 'width=200,height=' + (this.time_comp ? 215 : 190) +
                ',status=no,resizable=no,top=200,left=200,dependent=yes,alwaysRaised=yes'
        );
        obj_calwindow.opener = window;
        obj_calwindow.focus();
}

// timestamp generating function
function cal_gen_tsmp1 (dt_datetime) {
        return(this.gen_date(dt_datetime) + ' ' + this.gen_time(dt_datetime));
}

// date generating function
function cal_gen_date1 (dt_datetime) {
        return (
                (dt_datetime.getDate() < 10 ? '0' : '') + dt_datetime.getDate() + "/"
                + (dt_datetime.getMonth() < 9 ? '0' : '') + (dt_datetime.getMonth() + 1) + "/"
                + dt_datetime.getFullYear()
        );
}
// time generating function
function cal_gen_time1 (dt_datetime) {
        return (
                (dt_datetime.getHours() < 10 ? '0' : '') + dt_datetime.getHours() + ":"
                + (dt_datetime.getMinutes() < 10 ? '0' : '') + (dt_datetime.getMinutes()) + ":"
                + (dt_datetime.getSeconds() < 10 ? '0' : '') + (dt_datetime.getSeconds())
        );
}

// timestamp parsing function
function cal_prs_tsmp1 (str_datetime) {
        // if no parameter specified return current timestamp
```

```
	if (!str_datetime)
		return (new Date());

	// if positive integer treat as milliseconds from epoch
	if (RE_NUM.exec(str_datetime))
		return new Date(str_datetime);

	// else treat as date in string format
	var arr_datetime = str_datetime.split(' ');
	return this.prs_time(arr_datetime[1], this.prs_date(arr_datetime[0]));
}

// date parsing function
function cal_prs_date1 (str_date) {

	var arr_date = str_date.split('/');

	if (arr_date.length != 3) return cal_error ("Invalid date format: '" + str_date + "'.\nFormat
accepted is dd-mm-yyyy.");
	if (!arr_date[0]) return cal_error ("Invalid date format: '" + str_date + "'.\nNo day of
month value can be found.");
	if (!RE_NUM.exec(arr_date[0])) return cal_error ("Invalid day of month value: '" +
arr_date[0] + "'.\nAllowed values are unsigned integers.");
	if (!arr_date[1]) return cal_error ("Invalid date format: '" + str_date + "'.\nNo month value
can be found.");
	if (!RE_NUM.exec(arr_date[1])) return cal_error ("Invalid month value: '" + arr_date[1] +
"'.\nAllowed values are unsigned integers.");
	if (!arr_date[2]) return cal_error ("Invalid date format: '" + str_date + "'.\nNo year value
can be found.");
	if (!RE_NUM.exec(arr_date[2])) return cal_error ("Invalid year value: '" + arr_date[2] +
"'.\nAllowed values are unsigned integers.");

	var dt_date = new Date();
	dt_date.setDate(1);

	if (arr_date[1] < 1 || arr_date[1] > 12) return cal_error ("Invalid month value: '" +
arr_date[1] + "'.\nAllowed range is 01-12.");
	dt_date.setMonth(arr_date[1]-1);

	if (arr_date[2] < 100) arr_date[2] = Number(arr_date[2]) + (arr_date[2] <
NUM_CENTYEAR ? 2000 : 1900);
	dt_date.setFullYear(arr_date[2]);

	var dt_numdays = new Date(arr_date[2], arr_date[1], 0);
	dt_date.setDate(arr_date[0]);
	if (dt_date.getMonth() != (arr_date[1]-1)) return cal_error ("Invalid day of month value: '"
+ arr_date[0] + "'.\nAllowed range is 01-"+dt_numdays.getDate()+".");

	return (dt_date)
}

// time parsing function
function cal_prs_time1 (str_time, dt_date) {

	if (!dt_date) return null;
	var arr_time = String(str_time ? str_time : '').split(':');

	if (!arr_time[0]) dt_date.setHours(0);
	else if (RE_NUM.exec(arr_time[0]))
		if (arr_time[0] < 24) dt_date.setHours(arr_time[0]);
```

```
                else return cal_error ("Invalid hours value: '" + arr_time[0] + "'.\nAllowed range
is 00-23.");
        else return cal_error ("Invalid hours value: '" + arr_time[0] + "'.\nAllowed values are
unsigned integers.");

        if (!arr_time[1]) dt_date.setMinutes(0);
        else if (RE_NUM.exec(arr_time[1]))
                if (arr_time[1] < 60) dt_date.setMinutes(arr_time[1]);
                else return cal_error ("Invalid minutes value: '" + arr_time[1] + "'.\nAllowed
range is 00-59.");
        else return cal_error ("Invalid minutes value: '" + arr_time[1] + "'.\nAllowed values are
unsigned integers.");

        if (!arr_time[2]) dt_date.setSeconds(0);
        else if (RE_NUM.exec(arr_time[2]))
                if (arr_time[2] < 60) dt_date.setSeconds(arr_time[2]);
                else return cal_error ("Invalid seconds value: '" + arr_time[2] + "'.\nAllowed
range is 00-59.");
        else return cal_error ("Invalid seconds value: '" + arr_time[2] + "'.\nAllowed values are
unsigned integers.");

        dt_date.setMilliseconds(0);
        return dt_date;
}

function cal_error (str_message) {
        alert (str_message);
        return null;
}
```

Nome do arquivo: conselhoProfissional.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<select name="cons" >
        <option value="CRAS">CRAS</option>
        <option value="COREN">COREN</option>
        <option value="CRF">CRF</option>
        <option value="CRFA">CRFA</option>
        <option value="CREFITO">CREFITO</option>
        <option value="CRM">CRM</option>
        <option value="CRV">CRV</option>
        <option value="CRN">CRN</option>
        <option value="CRO">CRO</option>
        <option value="CRP">CRP</option>
        <option value="OUTROS">OUTROS</option>
</select>
```

Nome do arquivo: consultarGuia.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<!-- Javascript do Calendario -->
<script language="JavaScript" src="calendar1.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
<title>Consultar Guia</title>
</head>
<body>
        <c:if test="${!empty alerta}">
                <script language="JavaScript">
                        alert('${alerta}');
                </script>
        </c:if>
        <form id="formEditar" name="formConsultar" method="post"
action="ImprimeGuiaXML.ts">
                <input type=hidden name="operacao" value="consultar">
                <input type=hidden name="id" value="">
        </form>
        <form id="filtro" name="filtro" method="post" action="svtConsultarGuia.ts"
class="niceform">
        <h3> Consultar Guias Emitidas </h3>
        <fieldset><legend>Filtros de Pesquisa</legend>
        <div class="linha1"><label>Data inicial: <input name="datainicial" type="text"
id="datainicial"
        size="8" readonly="readonly" value="${datainicial}">
        <a href="javascript:cal1.popup();"><img src="img/cal.gif" width="16" height="16"
border="0" alt="Clique para selecionar a data"></a>
        <br>
        </label></div>
        <div class="linha1"><label>Data final: <input name="datafinal" type="text"
id="datafinal"
        size="8" readonly="readonly" value="${datafinal}">
        <a href="javascript:cal2.popup();"><img src="img/cal.gif" width="16" height="16"
border="0" alt="Clique para selecionar a data"></a>
        <br>
        </label></div>
        <div class="linha1"><label>Código da Operadora: <input name="codoperadora"
        type="text" id="codoperadora" size="15" maxlength="9"
value="${codoperadora}"></label>
        </div>
        <div class="linha1"><label>Guia
        <select name="ctguia">
        <option value=""> </option>
        <option value="ct_guiaConsulta">Consulta</option>
        <option value="ct_guiaSP_SADT">SP_SADT</option>
        <option value="ct_guiaSolicInternacao">Solicitação de Internação</option>
        <option value="ct_guiaResumoInternacao">Resumo de Internação</option>
        <option value="ct_guiaHonorarioIndividual">Honorário Individual</option>
        </select>
        </label></div>
        <div class="linha1"><input type="Submit" name="pesquisar" value="Pesquisar"
/></div></fieldset>
        </form>
        <c:if test="${!empty listaguias}">
                <table border="1">
                        <tr>
                                <th> Data Emissão </th>
                                <th> Operadora </th>
                                <th> Beneficiário </th>
                                <th> Hash </th>
                        </tr>
```

```
                        <c:forEach var="guias" items="${listaguias}">
                                <tr>
                                        <td>${guias.dataEmissao}</td>
                                        <td>${guias.clOperadora.nome}</td>
                                        <td>${guias.clBeneficiario.nome}</td>
        <td><a
href="SvtImprimeGuiaXML.ts?codope=${guias.clOperadora.codigo}&codbene=${guias.clBenef
iciario.codigo}&hash=${guias.hash}"
onclick="window.open('SvtImprimeGuiaXML.ts?codope=${guias.clOperadora.codigo}&codben
e=${guias.clBeneficiario.codigo}&hash=${guias.hash}'); return false;">${guias.hash}</a><td>
                                </tr>

                        </c:forEach>
                </table>
        </c:if>
<script>
        var cal1 = new calendar1(
        document.forms['filtro'].elements['datainicial']);
</script>
<script>
        var cal2 = new calendar1(
        document.forms['filtro'].elements['datafinal']);
</script>
</body>
</html>
```

Nome do arquivo: dtree.cs

```
.dtree {
        font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
        font-size: 11px;
        color: #666;
        white-space: nowrap;
}
.dtree img {
        border: 0px;
        vertical-align: middle;
}
.dtree a {
        color: #333;
        text-decoration: none;
}
.dtree a.node, .dtree a.nodeSel {
        white-space: nowrap;
        padding: 1px 2px 1px 2px;
}
.dtree a.node:hover, .dtree a.nodeSel:hover {
        color: #333;
        text-decoration: underline;
}
.dtree a.nodeSel {
        background-color: #c0d2ec;
}
.dtree .clip {
        overflow: hidden;
}
```

Nome do arquivo: dtree.js

```
// Node object
function Node(id, pid, name, url, title, target, icon, iconOpen, open) {
	this.id = id;
	this.pid = pid;
	this.name = name;
	this.url = url;
	this.title = title;
	this.target = target;
	this.icon = icon;
	this.iconOpen = iconOpen;
	this._io = open || false;
	this._is = false;
	this._ls = false;
	this._hc = false;
	this._ai = 0;
	this._p;
};

// Tree object
function dTree(objName) {
	this.config = {
		target				: null,
		folderLinks			: true,
		useSelection		: true,
		useCookies			: true,
		useLines			: true,
		useIcons			: true,
		useStatusText		: false,
		closeSameLevel		: false,
		inOrder				: false
```

```
	}
	this.icon = {
		root			: 'img/base.gif',
		folder			: 'img/folder.gif',
		folderOpen		: 'img/folderopen.gif',
		node			: 'img/page.gif',
		empty			: 'img/empty.gif',
		line			: 'img/line.gif',
		join			: 'img/join.gif',
		joinBottom		: 'img/joinbottom.gif',
		plus			: 'img/plus.gif',
		plusBottom		: 'img/plusbottom.gif',
		minus			: 'img/minus.gif',
		minusBottom		: 'img/minusbottom.gif',
		nlPlus			: 'img/nolines_plus.gif',
		nlMinus			: 'img/nolines_minus.gif'
	};
	this.obj = objName;
	this.aNodes = [];
	this.aIndent = [];
	this.root = new Node(-1);
	this.selectedNode = null;
	this.selectedFound = false;
	this.completed = false;
};

// Adds a new node to the node array
dTree.prototype.add = function(id, pid, name, url, title, target, icon, iconOpen, open) {
	this.aNodes[this.aNodes.length] = new Node(id, pid, name, url, title, target, icon,
iconOpen, open);
```

```
};

// Open/close all nodes
dTree.prototype.openAll = function() {
	this.oAll(true);
};
dTree.prototype.closeAll = function() {
	this.oAll(false);
};

// Outputs the tree to the page
dTree.prototype.toString = function() {
	var str = '<div class="dtree">\n';
	if (document.getElementById) {
		if (this.config.useCookies) this.selectedNode = this.getSelected();
		str += this.addNode(this.root);
	} else str += 'Browser not supported.';
	str += '</div>';
	if (!this.selectedFound) this.selectedNode = null;
	this.completed = true;
	return str;
};

// Creates the tree structure
dTree.prototype.addNode = function(pNode) {
	var str = '';
	var n=0;
	if (this.config.inOrder) n = pNode._ai;
	for (n; n<this.aNodes.length; n++) {
		if (this.aNodes[n].pid == pNode.id) {
```

```
                        var cn = this.aNodes[n];
                        cn._p = pNode;
                        cn._ai = n;
                        this.setCS(cn);
                        if (!cn.target && this.config.target) cn.target = this.config.target;
                        if (cn._hc && !cn._io && this.config.useCookies) cn._io =
this.isOpen(cn.id);
                        if (!this.config.folderLinks && cn._hc) cn.url = null;
                        if (this.config.useSelection && cn.id == this.selectedNode &&
!this.selectedFound) {
                                cn._is = true;
                                this.selectedNode = n;
                                this.selectedFound = true;
                        }
                        str += this.node(cn, n);
                        if (cn._ls) break;
                }
        }
        return str;
};

// Creates the node icon, url and text
dTree.prototype.node = function(node, nodeId) {
        var str = '<div class="dTreeNode">' + this.indent(node, nodeId);
        if (this.config.useIcons) {
                if (!node.icon) node.icon = (this.root.id == node.pid) ? this.icon.root :
((node._hc) ? this.icon.folder : this.icon.node);
                if (!node.iconOpen) node.iconOpen = (node._hc) ? this.icon.folderOpen :
this.icon.node;
                if (this.root.id == node.pid) {
                        node.icon = this.icon.root;
                        node.iconOpen = this.icon.root;
```

```
                }
                str += '<img id="i' + this.obj + nodeId + '" src="' + ((node._io) ? node.iconOpen :
node.icon) + '" alt="" />';
        }
        if (node.url) {
                str += '<a id="s' + this.obj + nodeId + '" class="' + ((this.config.useSelection) ?
((node._is ? 'nodeSel' : 'node')) : 'node') + '" href="' + node.url + '"';
                if (node.title) str += ' title="' + node.title + '"';
                if (node.target) str += ' target="' + node.target + '"';
                if (this.config.useStatusText) str += ' onmouseover="window.status=\'' +
node.name + '\';return true;" onmouseout="window.status=\'\';return true;" ';
                if (this.config.useSelection && ((node._hc && this.config.folderLinks) ||
!node._hc))
                        str += ' onclick="javascript: ' + this.obj + '.s(' + nodeId + ');"';
                str += '>';
        }
        else if ((!this.config.folderLinks || !node.url) && node._hc && node.pid != this.root.id)
                str += '<a href="javascript: ' + this.obj + '.o(' + nodeId + ');" class="node">';
        str += node.name;
        if (node.url || ((!this.config.folderLinks || !node.url) && node._hc)) str += '</a>';
        str += '</div>';
        if (node._hc) {
                str += '<div id="d' + this.obj + nodeId + '" class="clip" style="display:' +
((this.root.id == node.pid || node._io) ? 'block' : 'none') + ';">';
                str += this.addNode(node);
                str += '</div>';
        }
        this.aIndent.pop();
        return str;
};

// Adds the empty and line icons
dTree.prototype.indent = function(node, nodeId) {
```

```
	var str = '';

	if (this.root.id != node.pid) {

		for (var n=0; n<this.aIndent.length; n++)

			str += '<img src="' + ( (this.aIndent[n] == 1 && this.config.useLines) ?
this.icon.line : this.icon.empty ) + '" alt="" />';

		(node._ls) ? this.aIndent.push(0) : this.aIndent.push(1);

		if (node._hc) {

			str += '<a href="javascript: ' + this.obj + '.o(' + nodeId + ');"><img id="j'
+ this.obj + nodeId + '" src="';

			if (!this.config.useLines) str += (node._io) ? this.icon.nlMinus :
this.icon.nlPlus;

			else str += ( (node._io) ? ((node._ls && this.config.useLines) ?
this.icon.minusBottom : this.icon.minus) : ((node._ls && this.config.useLines) ?
this.icon.plusBottom : this.icon.plus ) );

			str += '" alt="" /></a>';

		} else str += '<img src="' + ( (this.config.useLines) ? ((node._ls) ?
this.icon.joinBottom : this.icon.join ) : this.icon.empty) + '" alt="" />';

	}

	return str;

};


// Checks if a node has any children and if it is the last sibling

dTree.prototype.setCS = function(node) {

	var lastId;

	for (var n=0; n<this.aNodes.length; n++) {

		if (this.aNodes[n].pid == node.id) node._hc = true;

		if (this.aNodes[n].pid == node.pid) lastId = this.aNodes[n].id;

	}

	if (lastId==node.id) node._ls = true;

};


// Returns the selected node

dTree.prototype.getSelected = function() {
```

```
        var sn = this.getCookie('cs' + this.obj);
        return (sn) ? sn : null;
};

// Highlights the selected node
dTree.prototype.s = function(id) {
        if (!this.config.useSelection) return;
        var cn = this.aNodes[id];
        if (cn._hc && !this.config.folderLinks) return;
        if (this.selectedNode != id) {
                if (this.selectedNode || this.selectedNode==0) {
                        eOld = document.getElementById("s" + this.obj + this.selectedNode);
                        eOld.className = "node";
                }
                eNew = document.getElementById("s" + this.obj + id);
                eNew.className = "nodeSel";
                this.selectedNode = id;
                if (this.config.useCookies) this.setCookie('cs' + this.obj, cn.id);
        }
};

// Toggle Open or close
dTree.prototype.o = function(id) {
        var cn = this.aNodes[id];
        this.nodeStatus(!cn._io, id, cn._ls);
        cn._io = !cn._io;
        if (this.config.closeSameLevel) this.closeLevel(cn);
        if (this.config.useCookies) this.updateCookie();
};
```

```
// Open or close all nodes
dTree.prototype.oAll = function(status) {
	for (var n=0; n<this.aNodes.length; n++) {
		if (this.aNodes[n]._hc && this.aNodes[n].pid != this.root.id) {
			this.nodeStatus(status, n, this.aNodes[n]._ls)
			this.aNodes[n]._io = status;
		}
	}
	if (this.config.useCookies) this.updateCookie();
};

// Opens the tree to a specific node
dTree.prototype.openTo = function(nId, bSelect, bFirst) {
	if (!bFirst) {
		for (var n=0; n<this.aNodes.length; n++) {
			if (this.aNodes[n].id == nId) {
				nId=n;
				break;
			}
		}
	}
	var cn=this.aNodes[nId];
	if (cn.pid==this.root.id || !cn._p) return;
	cn._io = true;
	cn._is = bSelect;
	if (this.completed && cn._hc) this.nodeStatus(true, cn._ai, cn._ls);
	if (this.completed && bSelect) this.s(cn._ai);
	else if (bSelect) this._sn=cn._ai;
	this.openTo(cn._p._ai, false, true);
};
```

```
// Closes all nodes on the same level as certain node
dTree.prototype.closeLevel = function(node) {
	for (var n=0; n<this.aNodes.length; n++) {
		if (this.aNodes[n].pid == node.pid && this.aNodes[n].id != node.id &&
this.aNodes[n]._hc) {
			this.nodeStatus(false, n, this.aNodes[n]._ls);
			this.aNodes[n]._io = false;
			this.closeAllChildren(this.aNodes[n]);
		}
	}
}

// Closes all children of a node
dTree.prototype.closeAllChildren = function(node) {
	for (var n=0; n<this.aNodes.length; n++) {
		if (this.aNodes[n].pid == node.id && this.aNodes[n]._hc) {
			if (this.aNodes[n]._io) this.nodeStatus(false, n, this.aNodes[n]._ls);
			this.aNodes[n]._io = false;
			this.closeAllChildren(this.aNodes[n]);
		}
	}
}

// Change the status of a node(open or closed)
dTree.prototype.nodeStatus = function(status, id, bottom) {
	eDiv	= document.getElementById('d' + this.obj + id);
	eJoin	= document.getElementById('j' + this.obj + id);
	if (this.config.useIcons) {
		eIcon	= document.getElementById('i' + this.obj + id);
```

```
            eIcon.src = (status) ? this.aNodes[id].iconOpen : this.aNodes[id].icon;
        }
        eJoin.src = (this.config.useLines)?
        ((status)?((bottom)?this.icon.minusBottom:this.icon.minus):((bottom)?this.icon.plusBott
om:this.icon.plus)):
        ((status)?this.icon.nlMinus:this.icon.nlPlus);
        eDiv.style.display = (status) ? 'block': 'none';
};


// [Cookie] Clears a cookie
dTree.prototype.clearCookie = function() {
        var now = new Date();
        var yesterday = new Date(now.getTime() - 1000 * 60 * 60 * 24);
        this.setCookie('co'+this.obj, 'cookieValue', yesterday);
        this.setCookie('cs'+this.obj, 'cookieValue', yesterday);
};


// [Cookie] Sets value in a cookie
dTree.prototype.setCookie = function(cookieName, cookieValue, expires, path, domain,
secure) {
        document.cookie =
                escape(cookieName) + '=' + escape(cookieValue)
                + (expires ? '; expires=' + expires.toGMTString() : '')
                + (path ? '; path=' + path : '')
                + (domain ? '; domain=' + domain : '')
                + (secure ? '; secure' : '');
};


// [Cookie] Gets a value from a cookie
dTree.prototype.getCookie = function(cookieName) {
```

```
	var cookieValue = '';

	var posName = document.cookie.indexOf(escape(cookieName) + '=');
	if (posName != -1) {

		var posValue = posName + (escape(cookieName) + '=').length;

		var endPos = document.cookie.indexOf(';', posValue);

		if (endPos != -1) cookieValue =
unescape(document.cookie.substring(posValue, endPos));

		else cookieValue = unescape(document.cookie.substring(posValue));

	}

	return (cookieValue);

};


// [Cookie] Returns ids of open nodes as a string

dTree.prototype.updateCookie = function() {

	var str = '';

	for (var n=0; n<this.aNodes.length; n++) {

		if (this.aNodes[n]._io && this.aNodes[n].pid != this.root.id) {

			if (str) str += '.';

			str += this.aNodes[n].id;

		}

	}

	this.setCookie('co' + this.obj, str);

};


// [Cookie] Checks if a node id is in a cookie

dTree.prototype.isOpen = function(id) {

	var aOpen = this.getCookie('co' + this.obj).split('.');

	for (var n=0; n<aOpen.length; n++)

		if (aOpen[n] == id) return true;

	return false;

};
```

```
// If Push and pop is not implemented by the browser
if (!Array.prototype.push) {
        Array.prototype.push = function array_push() {
                for(var i=0;i<arguments.length;i++)
                        this[this.length]=arguments[i];
                return this.length;
        }
};
if (!Array.prototype.pop) {
        Array.prototype.pop = function array_pop() {
                lastElement = this[this.length-1];
                this.length = Math.max(this.length-1,0);
                return lastElement;
        }
};
```

Nome do arquivo: escolheBen.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Beneficiário</title>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <form id="filtro" name="filtro" method="post"
action="SvtEscolheBeneficiario.ts?nomeoperadora=${nomeoperadora}" class="niceform">
        <input type="hidden" name="guia" value="${guia}">
        <input type="hidden" name="complex" value="${complex}">
        <input type="hidden" name="operadora" value="${operadora}">
        <h3>${tituloGuia}</h3>
        <div class="linha"><label>Operadora: <input name="nomeoperadora"
                type="text" id="nomeoperadora" size="70" value="${nomeoperadora}"
disabled="disabled"> </label>
        </div>
        <br>
        <fieldset><legend>Busca Beneficiario</legend>
        <div class="linha"><label>Nome: <input name="nome"
```

```
                type="text" id="nome" size="70" value="${nome}"> </label>
        </div>
        <input type="Submit" name="pesquisar" value="Pesquisar" /></fieldset>
        </form>
        <c:if test="${!empty listaBeneficiarios}">
                <table border="1">
                        <tr>
                                <th>Nome</th>
                                <th>Número da Carteira</th>
                        </tr>
                        <c:forEach var="beneficiario" items="${listaBeneficiarios}">
                                <tr>
                                        <td>${beneficiario.nome}</td>
                                        <td><a
href="SvtEscolhePrestador.ts?guia=${guia}&complex=${complex}&operadora=${operadora}&b
eneficiario=${beneficiario.codigo}&nomeoperadora=${nomeoperadora}&nomebene=${benefici
ario.nome}">${beneficiario.numcarteira}</a></td>
                                </tr>
                        </c:forEach>
                </table>
        </c:if>
</body>
</html>
```

Nome do arquivo: escolheCon.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <form id="filtro" name="filtro" method="post"
action="SvtEscolhePrestador.ts?nomeoperadora=${nomeoperadora}&nomebene=${nomebene
}" class="niceform">
        <input type="hidden" name="guia" value="${guia}">
        <input type="hidden" name="complex" value="${complex}">
        <input type="hidden" name="operadora" value="${operadora}">
        <input type="hidden" name="beneficiario" value="${beneficiario}">
        <h3>${tituloGuia}</h3>
        <div class="linha"><label>Operadora: <input name="nomeoperadora"
                type="text" id="nomeoperadora" size="70" value="${nomeoperadora}"
disabled="disabled"> </label>
        </div>
        <div class="linha"><label>Beneficiário: <input name="nomebene"
                type="text" id="nomebene" size="70" value="${nomebene}"
disabled="disabled"> </label>
        </div>
        <br>
        <fieldset><legend>Busca Contratado Principal</legend>
        <div class="linha"><label>Nome: <input name="nome"
                type="text" id="nome" size="70" value="${nome}"> </label>
        </div>
```

```
        <input type="Submit" name="pesquisar" value="Pesquisar" /></fieldset>
        </form>
        <c:if test="${!empty listaPrestadores}">
                <table border="1">
                        <tr>
                                <th>CNPJ/CPF</th>
                                <th>Nome</th>
                        </tr>
                        <c:forEach var="prestador" items="${listaPrestadores}">
                                <tr>
                                        <c:if test="${prestador.tipo=='fisico'}">
                                        <td>${prestador.cpf}</td>
                                        <td><a
href="SvtPrePreenchimentoGuia.ts?guia=${guia}&complex=${complex}&operadora=${operado
ra}&beneficiario=${beneficiario}&contratado=${prestador.cpf}">${prestador.nome}</a></td>
                                        </c:if>
                                        <c:if test="${prestador.tipo=='juridico'}">
                                        <td>${prestador.cnpj}</td>
                                        <td><a
href="SvtPrePreenchimentoGuia.ts?guia=${guia}&complex=${complex}&operadora=${operado
ra}&beneficiario=${beneficiario}&contratado=${prestador.cnpj}">${prestador.nome}</a></td>
                                        </c:if>
                                </tr>
                        </c:forEach>
                </table>
        </c:if>
</body>
</html>
```

Nome do arquivo: escolheOpe.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Operadora</title>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <form id="filtro" name="filtro" method="post" action="SvtEscolheOperadora.ts"
class="niceform">
        <input type="hidden" name="guia" value="${guia}">
        <input type="hidden" name="complex" value="${complex}">
        <h3>${tituloGuia}</h3>
        <fieldset><legend>Busca Operadora</legend>
        <div class="linha"><label>Nome: <input name="nome"
                type="text" id="nome" size="70" value="${nome}"> </label>
        </div>
        <input type="Submit" name="pesquisar" value="Pesquisar" /></fieldset>
        </form>
        <c:if test="${!empty listaOperadoras}">
                <table border="1">
                        <tr>
                                <th>Registro</th>
                                <th>Nome Operadora</th>
```

```
                        </tr>
                        <c:forEach var="operadora" items="${listaOperadoras}">
                                <tr>
                                        <td>${operadora.registro}</td>
                                        <td><a
href="SvtEscolheBeneficiario.ts?guia=${guia}&complex=${complex}&operadora=${operadora.c
odigo}&nomeoperadora=${operadora.nome}">${operadora.nome}</a></td>
                                </tr>
                        </c:forEach>
                </table>
        </c:if>
</body>
</html>
```

Nome do arquivo: formulario.css

```
form
{
        border: solid black 1px;
}
form fieldset
{
  float: left;
  margin-right: 1px;
  width: 80%;
  border: solid black 1px;
  padding: 3%;
  margin-bottom: 1px;
}

form legend
{
  padding: 0px;
  margin: 0px;
  border: solid black 1px;
  font-size: 60%;
  font-weight: bold;
  background-color: #e8e8e8;
}

form label
{
  font-size: 10px;
}

form input
{
  size: 15px;

  border: solid #ccc 1px;
  font-size: 11px;
  font-family: 'Trebuchet MS', Verdana, Tahoma, Serif;
}

form input.botao
{
  display: block;
  width: auto;
  float: right;
```

```
}
```

Nome do arquivo: guiaConsulta.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="js/funcoesGuia.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Guia Consulta para impressão</title>

</head>
<body>
<img src="img/consulta.PNG" style="display:block; position:absolute; z-index:5; left: 0px; top:
0px;">
<div id="template" style="display:block; background-color:transparent; z-index:6; position:
absolute; left: 0px; top: 0px;"></div>
<script>
<!-- campos -->

inserirEmCamposNumericos('${guia.STRnumeroCarteira}', 79, 324);
inserirEmCamposData('${guia.STRdataEmissaoGuia}', 314, 251);
inserirEmCamposData('${guia.STRvalidadeCarteira}', 897, 323);
inserirEmCamposNumericos('${guia.STRnumeroCNS}', 713, 380);
inserirEmCamposNumericos('${guia.STRCNPJ}${guia.STRCPF}${guia.STRcodigoPrestadorN
aOperadora}', 80, 461);
inserirEmCamposData('${guia.STRdataAtendimento}', 75, 771);
inserirEmCamposNumericos('${guia.STRcodigoTabela}', 315, 772);
inserirEmCamposNumericos('${guia.STRcodigoProcedimento}', 412,772);
</script>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
70px; top: 120px;"><img src="img/logo1.JPG"></div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
879px; top: 168px;">${guia.STRnumeroGuiaPrestador}
${guia.STRnumeroGuiaOperadora}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
75px; top: 255px;">${guia.STRregistroANS}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
568px; top: 325px;">${guia.STRnomePlano}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
70px; top: 387px;">${guia.STRnomeBeneficiario}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
433px; top: 469px;">${guia.STRnomeContratado}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
973px; top: 472px;">${guia.STRnumeroCNES}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
897px; top: 577px;">${guia.STRcodigoUF}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
74px; top: 524px;">${guia.STRtipoLogradouro}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
132px; top: 526px;">${guia.STRlogradouro},
${guia.STRnumero},${guia.STRcomplemento}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
556px; top: 525px;">${guia.STRmunicipio}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
796px; top: 525px;">${guia.STRcodigoUF}</div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
861px; top: 525px;">${guia.STRcodigoIBGEMunicipio}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
986px; top: 525px;">${guia.STRcep}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
70px; top: 581px;">${guia.STRnomeProfissional}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
558px; top: 582px;">${guia.STRsiglaConselho}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
738px; top: 582px;">${guia.STRnumeroConselho}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
967px; top: 583px;">${guia.STRcbos}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
76px; top: 646px;">${guia.STtipoDoenca}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
238px; top: 646px;">${guia.STunidadeTempo}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
481px; top: 647px;">${guia.STRindicadorAcidente}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
75px; top: 687px;">${guia.STcodigoDiagnostico}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
211px; top: 687px;">${guia.STcodigoDiagnosticoSec[0]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
344px; top: 687px;">${guia.STcodigoDiagnosticoSec[1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
478px; top: 700px;">${guia.STcodigoDiagnosticoSec[2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
77px; top: 827px;">${guia.STRtipoConsulta}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
409px; top: 828px;">${guia.STRtipoSaida}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
76px; top: 875px; width: 960px; line-height: 23pt">${guia.STRobservacao}</div>

</body>
</html>
```

Nome do arquivo: guiaHonorarioIndividual.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="js/funcoesGuia.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Guia Honorário Individual para impressão</title>
</head>
<body>
<img src="img/guiaHonorarioIndividual.png" style="display:block; position:absolute; z-index:5;
left: 0px; top: 0px;">
<div id="template" style="display:block; background-color:transparent; z-index:6; position:
absolute; left: 0px; top: 0px;"></div>
<script>
<!-- campos -->
inserirEmCamposNumericos('${guia.STRnumeroGuiaPrincipal}', 260, 184);
inserirEmCamposData('${guia.STRdataEmissaoGuia}',717, 186);
inserirEmCamposNumericos('${guia.STRnumeroCarteira}',52, 251);
inserirEmCamposData('${guia.STRvalidadeCarteira}',675, 251);
```

```
inserirEmCamposNumericos('${guia.STRnumeroCNS}', 704, 305);
inserirEmCamposNumericos('${guia.STRCNPJ}${guia.STRCPF}${guia.STRcodigoPrestadorN
aOperadora}', 54, 365);
inserirEmCamposNumericos('${guia.STRCNPJExe}${guia.STRCPFExe}${guia.STRcodigoPres
tadorNaOperadoraExe}', 52, 428);
inserirEmCamposNumericos('${guia.STRtipoAcomodacaoExe}', 976, 427);
inserirEmCamposNumericos('${guia.STRposicaoProfissionalExe}', 56, 472);
inserirEmCamposData('${guia.STRdata[pagina*10]}', 64, 544);
inserirEmCamposData('${guia.STRdata[pagina*10+1]}', 64, 576);
inserirEmCamposData('${guia.STRdata[pagina*10+2]}', 64, 608);
inserirEmCamposData('${guia.STRdata[pagina*10+3]}', 64, 640);
inserirEmCamposData('${guia.STRdata[pagina*10+4]}', 64, 670);
inserirEmCamposData('${guia.STRdata[pagina*10+5]}', 64, 702);
inserirEmCamposData('${guia.STRdata[pagina*10+6]}', 64, 732);
inserirEmCamposData('${guia.STRdata[pagina*10+7]}', 64, 762);
inserirEmCamposData('${guia.STRdata[pagina*10+8]}', 64, 792);
inserirEmCamposData('${guia.STRdata[pagina*10+9]}', 64, 821);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10]}', 228, 545);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+1]}', 228, 576);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+2]}', 228, 608);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+3]}', 228, 640);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+4]}', 228, 670);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+5]}', 228, 702);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+6]}', 228, 732);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+7]}', 228, 762);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+8]}', 228, 792);
inserirEmCamposHora('${guia.STRhoraInicio[pagina*10+9]}', 228, 821);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10]}', 340, 544);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+1]}', 340, 576);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+2]}', 340, 608);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+3]}', 340, 640);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+4]}', 340, 670);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+5]}', 340, 702);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+6]}', 340, 732);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+7]}', 340, 762);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+8]}', 340, 792);
inserirEmCamposHora('${guia.STRhoraFim[pagina*10+9]}', 340, 821);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10]}', 451, 544);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+1]}', 451, 576);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+2]}', 451, 608);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+3]}', 451, 640);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+4]}', 451, 670);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+5]}', 451, 702);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+6]}', 451, 732);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+7]}', 451, 762);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+8]}', 451, 792);
inserirEmCamposNumericos('${guia.STRtipoTabela[pagina*10+9]}', 451, 821);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10]}', 507, 544);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+1]}', 507, 576);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+2]}', 507, 608);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+3]}', 507, 640);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+4]}', 507, 670);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+5]}', 507, 702);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+6]}', 507, 732);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+7]}', 507, 762);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+8]}', 507, 792);
inserirEmCamposNumericos('${guia.STRcodigo[pagina*10+9]}', 507, 821);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10]}', 986, 544);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+1]}', 986, 576);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+2]}', 986, 608);
```

```
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+3]}', 986, 640);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+4]}', 986, 670);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+5]}', 986, 702);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+6]}', 986, 732);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+7]}', 986, 762);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+8]}', 986, 792);
inserirEmCamposNumericos('${guia.STRquantidadeRealizada[pagina*10+9]}', 986, 821);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10]}', 1045, 544);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+1]}', 1045, 576);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+2]}', 1045, 608);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+3]}', 1045, 640);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+4]}', 1045, 670);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+5]}', 1045, 702);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+6]}', 1045, 732);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+7]}', 1045, 762);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+8]}', 1045, 792);
inserirEmCamposNumericos('${guia.STRviaAcesso[pagina*10+9]}', 1045, 821);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10]}', 1079, 544);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+1]}', 1079, 576);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+2]}', 1079, 608);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+3]}', 1079, 640);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+4]}', 1079, 670);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+5]}', 1079, 702);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+6]}', 1079, 732);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+7]}', 1079, 762);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+8]}', 1079, 792);
inserirEmCamposNumericos('${guia.STRtecnicaUtilizada[pagina*10+9]}', 1079, 821);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10]}', 1113, 544);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+1]}', 1113, 576);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+2]}', 1113, 608);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+3]}', 1113, 640);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+4]}', 1113, 670);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+5]}', 1113, 702);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+6]}', 1113, 732);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+7]}', 1113, 762);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+8]}', 1113, 792);
inserirEmCamposNumericos('${guia.STRreducaoAcrescimo[pagina*10+9]}', 1113, 821);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10]}', 1247, 544);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+1]}', 1247, 576);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+2]}', 1247, 608);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+3]}', 1247, 640);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+4]}', 1247, 670);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+5]}', 1247, 702);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+6]}', 1247, 732);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+7]}', 1247, 762);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+8]}', 1247, 792);
inserirEmCamposNumericos('${guia.STRvalor[pagina*10+9]}', 1247, 821);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10]}', 1425, 545);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+1]}', 1425, 576);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+2]}', 1425, 608);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+3]}', 1425, 640);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+4]}', 1425, 670);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+5]}', 1425, 702);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+6]}', 1425, 732);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+7]}', 1425, 762);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+8]}', 1425, 792);
inserirEmCamposNumericos('${guia.STRvalorTotal[pagina*10+9]}', 1425, 821);
inserirEmCAmposNumericos('${guia.STRtotalGeralHonorario}', 1327, 872);

</script>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
60px; top: 55px;"><img src="img/logo1.JPG"></div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
46px; top: 190px;">${guia.STRregistroANS}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1284px; top: 98px;">${guia.STRnumeroGuiaPrestador}
${guia.STRnumeroGuiaOperadora}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
505px; top: 257px;">${guia.STRnomePlano}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
44px; top: 304px;">${guia.STRnomeBeneficiario}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
378px; top: 369px;">${guia.STRnomeContratado}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1030px; top: 368px;">${guia.STRnumeroCNES}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
376px; top: 431px;">${guia.STRnomeContratadoExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
863px; top: 431px;">${guia.STRnumeroCNESExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
130px; top: 477px;">${guia.STRnomeExecutanteExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
697px; top: 477px;">${guia.STRsiglaConselhoExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
865px; top: 479px;">${guia.STRnumeroConselhoExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1028px; top: 478px;">${guia.STRufConselhoExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1097px; top: 479px;">${guia.STRCPFExe}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
746px; top: 544px;">${guia.STRdescricao[pagina*10]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
745px; top: 576px;">${guia.STRdescricao[pagina*10+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
747px; top: 608px;">${guia.STRdescricao[pagina*10+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
746px; top: 640px;">${guia.STRdescricao[pagina*10+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
747px; top: 670px;">${guia.STRdescricao[pagina*10+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
746px; top: 702px;">${guia.STRdescricao[pagina*10+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
744px; top: 733px;">${guia.STRdescricao[pagina*10+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
746px; top: 762px;">${guia.STRdescricao[pagina*10+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
746px; top: 793px;">${guia.STRdescricao[pagina*10+8]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
748px; top: 821px;">${guia.STRdescricao[pagina*10+9]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
49px; top: 915px;">${guia.STRobservacao}</div>

</body>
</html>
```

Nome do arquivo: guiaResumoInternacao.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="js/funcoesGuia.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Guia Solicitação de Internação para impressão</title>

</head>
<body>

<img src="img/guiaResumoInternacao.png" style="display:block; position:absolute; z-index:5;
left: 0px; top: 0px; page-break-after:always;">
<div id="template" style="display:block; background-color:transparent; z-index:6; position:
absolute; left: 0px; top: 0px;"></div>

<script>
<!-- campos -->
inserirEmCamposNumericos2('${guia.STRnumeroGuiaPrincipal}', 228,142);
inserirEmCamposData2('${guia.STRdataAutorizacao}', 625, 142);
inserirEmCamposData2('${guia.STRvalidadeSenha}', 911, 142);
inserirEmCamposData2('${guia.STRdataEmissaoGuia}', 1073, 142);
inserirEmCamposNumericos2('${guia.STRnumeroCarteira}', 47, 195);
inserirEmCamposData2('${guia.STRvalidadeCarteira}', 596, 197);
inserirEmCamposNumericos2('${guia.STRnumeroCNS}', 625, 239);
inserirEmCamposNumericos2('${guia.STRCNPJ}${guia.STRCPF}${guia.STRcodigoPrestador
NaOperadora}',51, 296 );
inserirEmCamposNumericos2('${guia.STRacomodacao}', 263, 393);
inserirEmCamposData2('${guia.STRdataHoraInternacao}', 443, 396);
inserirEmCamposData2('${guia.STRdataHoraSaidaInternacao}', 685, 396);
inserirEmCamposNumericos2('${guia.STRqtdNascidosVivosTermo}', 1067, 477);
inserirEmCamposNumericos2('${guia.STRqtdNascidosMortos}', 1218, 479);
inserirEmCamposNumericos2('${guia.STRqtdVivosPrematuros}', 1348, 479);
inserirEmCamposNumericos2('${guia.STRnomeTabela}', 51, 540);
inserirEmCamposNumericos2('${guia.STRindicadorAcidente}', 482, 541);
inserirEmCamposNumericos2('${guia.STRnomeTabelaOb}', 966, 540);
inserirEmCamposNumericos2('${guia.STRnumeroDeclaracaoOb}', 1078, 537);
inserirEmCamposData2('${guia.STRdataPR[pagina*14]}', 54, 598);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+1]}', 54, 622);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+2]}', 54, 648);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+3]}', 54, 672);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+4]}', 54, 698);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+5]}', 81,1195);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+6]}', 81,1221);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+7]}', 81,1248);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+8]}', 81,1275);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+9]}', 81,1301);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+10]}', 81,1326);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+11]}', 81,1354);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+12]}', 81,1378);
inserirEmCamposData2('${guia.STRdataPR[pagina*14+13]}', 81,1404);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14]}', 192, 598);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+1]}', 192, 622);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+2]}', 192, 648);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+3]}', 192, 672);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+4]}', 192, 698);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+5]}', 217, 1195);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+6]}', 217, 1221);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+7]}', 217, 1248);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+8]}', 217, 1275);
```

```
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+9]}', 217, 1301);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+10]}', 217, 1326);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+11]}', 217, 1354);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+12]}', 217, 1378);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*14+13]}', 217, 1404);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14]}', 281, 599);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+1]}', 281, 622);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+2]}', 281, 648);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+3]}', 281, 672);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+4]}', 281, 698);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+5]}', 308, 1195);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+6]}', 308, 1221);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+7]}', 308, 1248);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+8]}', 308, 1275);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+9]}', 308, 1301);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+10]}', 308, 1326);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+11]}', 308, 1354);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+12]}', 308, 1378);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*14+13]}', 308, 1404);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14]}', 376, 599);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+1]}', 376, 622);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+2]}', 376, 648);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+3]}', 376, 672);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+4]}', 376, 698);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+5]}', 403, 1195);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+6]}', 403, 1221);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+7]}', 403, 1248);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+8]}', 403, 1275);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+9]}', 403, 1301);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+10]}', 403, 1326);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+11]}', 403, 1354);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+12]}', 403, 1378);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*14+13]}', 403, 1404);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14]}', 426, 599);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+1]}', 426, 622);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+2]}', 426, 648);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+3]}', 426, 672);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+4]}', 426, 698);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+5]}', 452, 1195);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+6]}', 452, 1221);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+7]}', 452, 1248);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+8]}', 452, 1275);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+9]}', 452, 1301);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+10]}', 452, 1326);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+11]}', 452, 1354);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+12]}', 452, 1378);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*14+13]}', 452, 1404);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14]}',943,599);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+1]}',943,622);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+2]}',943,648);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+3]}',943,672);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+4]}',943,698);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+5]}',971,1195);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+6]}',971,1221);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+7]}',971,1248);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+8]}',971,1275);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+9]}',971,1301);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+10]}',971,1326);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+11]}',971,1354);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+12]}',971,1378);
```

```
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*14+13]}',971,1404);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14]}',1058 , 599);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+1]}',1058 , 622);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+2]}',1058 , 648);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+3]}',1058 , 672);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+4]}',1058 , 698);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+5]}',1085 , 1195);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+6]}',1085 , 1221);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+7]}',1085 , 1248);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+8]}',1085 , 1275);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+9]}',1085 , 1301);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+10]}',1085 , 1326);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+11]}',1085 , 1354);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+12]}',1085 , 1378);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*14+13]}',1085 , 1404);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14]}', 1171, 599);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+1]}', 1171, 622);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+2]}', 1171, 648);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+3]}', 1171, 672);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+4]}', 1171, 698);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+5]}', 1194, 1195);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+6]}', 1194, 1221);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+7]}', 1194, 1248);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+8]}', 1194,.1275);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+9]}', 1194, 1301);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+10]}', 1194, 1326);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+11]}', 1194, 1354);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+12]}', 1194, 1378);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*14+13]}', 1194, 1404);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14]}',1315, 599 );
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+1]}',1315, 622);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+2]}',1315, 648);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+3]}',1315, 672);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+4]}',1315, 698);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+5]}',1344, 1195);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+6]}',1344, 1221);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+7]}',1344, 1248);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+8]}',1344, 1275);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+9]}',1344, 1301);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+10]}',1344, 1326);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+11]}',1344, 1354);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+12]}',1344, 1378);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*14+13]}',1344, 1404);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14]}', 995, 599);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+1]}', 995, 622);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+2]}', 995, 648);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+3]}', 995, 672);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+4]}', 995, 698);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+5]}', 1020, 1195);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+6]}', 1020, 1221);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+7]}', 1020, 1248);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+8]}', 1020, 1275);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+9]}', 1020, 1301);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+10]}', 1020, 1326);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+11]}', 1020, 1354);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+12]}', 1020, 1378);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*14+13]}', 1020, 1404);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14]}', 1025, 599);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+1]}', 1025, 622);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+2]}', 1025, 648);
```

```
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+3]}', 1025, 672);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+4]}', 1025, 698);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+5]}', 1052, 1195);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+6]}', 1052, 1221);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+7]}', 1052, 1248);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+8]}', 1052, 1275);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+9]}', 1052, 1301);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+10]}', 1052, 1326);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+11]}', 1052, 1354);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+12]}', 1052, 1378);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*14+13]}', 1052, 1404);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*20]}', 108, 758);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*+1]}', 108, 782);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*2]}', 108, 807);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*3]}', 108, 832);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*4]}', 108, 858);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*5]}', 108, 882);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*6]}', 114, 1463);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*7]}', 114, 1488);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*8]}', 114, 1515);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*9]}', 114, 1541);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*10]}', 114, 1568);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*11]}', 114, 1594);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*12]}', 114, 1620);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*13]}', 114, 1646);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*14]}', 114, 1672);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*15]}', 114, 1700);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*16]}', 114, 1726);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*17]}', 114, 1751);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*18]}', 114, 1777);
inserirEmCamposNumericos2('${guia.STRposicaoProfissionalID[pagina*19]}', 114, 1805);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20]}${guia.
STRcpfPR[pagina*20]}', 174, 757);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+1]}${gu
ia.STRcpfPR[pagina*20+1]}', 174, 782);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+2]}${gu
ia.STRcpfPR[pagina*20+2]}', 174, 807);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+3]}${gu
ia.STRcpfPR[pagina*20+3]}', 174, 832);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+4]}${gu
ia.STRcpfPR[pagina*20+4]}', 174, 858);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+5]}${gu
ia.STRcpfPR[pagina*20+5]}', 174, 882);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+6]}${gu
ia.STRcpfPR[pagina*20+6]}', 179, 1463);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+7]}${gu
ia.STRcpfPR[pagina*20+7]}', 179, 1488);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+8]}${gu
ia.STRcpfPR[pagina*20+8]}', 179, 1515);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+9]}${gu
ia.STRcpfPR[pagina*20+9]}', 179, 1541);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+10]}${g
uia.STRcpfPR[pagina*20+10]}', 179, 1568);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+11]}${g
uia.STRcpfPR[pagina*20+11]}', 179, 1594);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+12]}${g
uia.STRcpfPR[pagina*20+12]}', 179, 1620);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+13]}${g
uia.STRcpfPR[pagina*20+13]}', 179, 1646);
```

```
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+14]}${g
uia.STRcpfPR[pagina*20+14]}', 179, 1672);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+15]}${g
uia.STRcpfPR[pagina*20+15]}', 179, 1700);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+16]}${g
uia.STRcpfPR[pagina*20+16]}', 179, 1726);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+17]}${g
uia.STRcpfPR[pagina*20+17]}', 179, 1751);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+18]}${g
uia.STRcpfPR[pagina*20+18]}', 179, 1777);
inserirEmCamposNumericos2('${guia.STRcodigoPrestadorNaOperadoraPR[pagina*20+19]}${g
uia.STRcpfPR[pagina*20+19]}', 179, 1805);
inserirEmCamposNumericos2('${guia.STRservicosExecutados}',193, 928 );
inserirEmCamposNumericos2('${guia.STRdiarias}', 378, 926);
inserirEmCamposNumericos2('${guia.STRtaxas}', 564, 928);
inserirEmCamposNumericos2('${guia.STRmateriais}', 747, 928);
inserirEmCamposNumericos2('${guia.STRmedicamentos}', 935, 929);
inserirEmCamposNumericos2('${guia.STRgases}', 1124, 929);
inserirEmCamposNumericos2('${guia.STRtotalGeral}', 1309, 929);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*5]}', 73, 1867);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*5+1]}', 73, 1893);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*5+2]}', 73, 1919);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*5+3]}', 73, 1944);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*5+4]}', 73, 1972);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*5]}', 123, 1867);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*5+1]}', 123, 1893);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*5+2]}', 123, 1919);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*5+3]}', 123, 1944);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*5+4]}', 123, 1972);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*5]}', 857, 1867);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*5+1]}', 857, 1894);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*5+2]}', 857, 1919);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*5+3]}', 857, 1944);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*5+4]}', 857, 1972);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[]pagina*5}', 1143, 1868);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[]pagina*5+1}', 1143, 1894);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[]pagina*5+2}', 1143, 1919);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[]pagina*5+3}', 1143, 1944);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[]pagina*5+4}', 1143, 1972);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[]pagina*5}', 1315, 1868);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[]pagina*5+1}', 1315, 1894);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[]pagina*5+2}', 1315, 1919);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[]pagina*5+3}', 1315, 1944);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[]pagina*5+4}', 1315, 1972);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPMOPM}', 1338, 2019);
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
908px; top: 1867px;">${guia.STRcodigoBarraOPM[pagina*5]}</div>


</script>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
60px; top: 50px;"><img src="img/logo1.JPG" height="65" width="120"></div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1263px; top: 71px;">${guia.STRnumeroGuiaPrestador}
${guia.STRnumeroGuiaOperadora}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
44px; top: 143px;">${guia.STRregistroANS}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
771px; top: 143px;">${guia.STRsenhaAutorizacao}</div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
450px; top: 198px;">${guia.STRnomePlano}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
42px; top: 242px;">${guia.STRnomeBeneficiario}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
332px; top: 297px;">${guia.STRnomeContratado}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
795px; top: 298px;">${guia.STRnumeroCNES}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
42px; top: 342px;">${guia.STRtipoLogradouro}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
94px; top: 341px;">${guia.STRlogradouro}, ${guia.STRnumero},
${guia.STRcomplemento}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
466px; top: 341px;">${guia.STRmunicipio}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
712px; top: 342px;">${guia.STRcodigoUF}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
761px; top: 342px;">${guia.STRcodigoIBGEMunicipio}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
838px; top: 341px;">${guia.STRcep}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
50px; top: 396px;">${guia.STRcaraterInternacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
926px; top: 396px;">${guia.STRtipoInternacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1260px; top: 397px;">${guia.STRregimeInternacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
49px; top: 437px;">${guia.STRemGestacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
134px; top: 437px;">${guia.STRaborto}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
194px; top: 437px;">${guia.STRtranstornoMaternoRelGravidez}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
418px; top: 438px;">${guia.STRcomplicacaoPeriodoPuerperio}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
538px; top: 438px;">${guia.STRatendimentoRNSalaParto}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
708px; top: 437px;">${guia.STRcomplicacaoNeonatal}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
844px; top: 438px;">${guia.STRbaixoPeso}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
961px; top: 437px;">${guia.STRpartoCesareo}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1067px; top: 438px;">${guia.STRpartoNormal}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
49px; top: 478px;">${guia.STRobitoMulher}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
565px; top: 478px;">${guia.STRqtdeobitoPrecoce} ${guia.STRqtdeobitoTardio}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
899px; top: 482px;">${guia.STRnumeroDN}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
625px; top: 599px;">${guia.STRdescricaoPR[pagina*14]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
625px; top: 622px;">${guia.STRdescricaoPR[pagina*14+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
625px; top: 648px;">${guia.STRdescricaoPR[pagina*14+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
625px; top: 672px;">${guia.STRdescricaoPR[pagina*14+3]}</div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
625px; top: 698px;">${guia.STRdescricaoPR[pagina*14+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1195px;">${guia.STRdescricaoPR[pagina*14+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1221px;">${guia.STRdescricaoPR[pagina*14+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1248px;">${guia.STRdescricaoPR[pagina*14+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1275px;">${guia.STRdescricaoPR[pagina*14+8]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1301px;">${guia.STRdescricaoPR[pagina*14+9]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1326px;">${guia.STRdescricaoPR[pagina*14+10]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1354px;">${guia.STRdescricaoPR[pagina*14+11]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1378px;">${guia.STRdescricaoPR[pagina*14+12]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
649px; top: 1404px;">${guia.STRdescricaoPR[pagina*14+13]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
465px; top: 758px;">${guia.STRnomeExecutanteID[pagina*20]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
465px; top: 782px;">${guia.STRnomeExecutanteID[pagina*20+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
465px; top: 807px;">${guia.STRnomeExecutanteID[pagina*20+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
465px; top: 832px;">${guia.STRnomeExecutanteID[pagina*20+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
465px; top: 858px;">${guia.STRnomeExecutanteID[pagina*20+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
465px; top: 882px;">${guia.STRnomeExecutanteID[pagina*20+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1463px;">${guia.STRnomeExecutanteID[pagina*20+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1489px;">${guia.STRnomeExecutanteID[pagina*20+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1515px;">${guia.STRnomeExecutanteID[pagina*20+8]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1541px;">${guia.STRnomeExecutanteID[pagina*20+9]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1567px;">${guia.STRnomeExecutanteID[pagina*20+10]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1593px;">${guia.STRnomeExecutanteID[pagina*20+11]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1621px;">${guia.STRnomeExecutanteID[pagina*20+12]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1647px;">${guia.STRnomeExecutanteID[pagina*20+13]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1673px;">${guia.STRnomeExecutanteID[pagina*20+14]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1700px;">${guia.STRnomeExecutanteID[pagina*20+15]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1726px;">${guia.STRnomeExecutanteID[pagina*20+16]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1751px;">${guia.STRnomeExecutanteID[pagina*20+17]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1777px;">${guia.STRnomeExecutanteID[pagina*20+18]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
469px; top: 1804px;">${guia.STRnomeExecutanteID[pagina*20+19]}</div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
854px; top: 758px;">${guia.STRsiglaConselhoID[pagina*20]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
854px; top: 782px;">${guia.STRsiglaConselhoID[pagina*20+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
854px; top: 807px;">${guia.STRsiglaConselhoID[pagina*20+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
854px; top: 832px;">${guia.STRsiglaConselhoID[pagina*20+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
854px; top: 858px;">${guia.STRsiglaConselhoID[pagina*20+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
854px; top: 882px;">${guia.STRsiglaConselhoID[pagina*20+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1463px;">${guia.STRsiglaConselhoID[pagina*20+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1489px;">${guia.STRsiglaConselhoID[pagina*20+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1515px;">${guia.STRsiglaConselhoID[pagina*20+8]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1541px;">${guia.STRsiglaConselhoID[pagina*20+9]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1567px;">${guia.STRsiglaConselhoID[pagina*20+10]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1593px;">${guia.STRsiglaConselhoID[pagina*20+11]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1621px;">${guia.STRsiglaConselhoID[pagina*20+12]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1647px;">${guia.STRsiglaConselhoID[pagina*20+13]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1673px;">${guia.STRsiglaConselhoID[pagina*20+14]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1700px;">${guia.STRsiglaConselhoID[pagina*20+15]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1726px;">${guia.STRsiglaConselhoID[pagina*20+16]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1751px;">${guia.STRsiglaConselhoID[pagina*20+17]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1777px;">${guia.STRsiglaConselhoID[pagina*20+18]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
813px; top: 1804px;">${guia.STRsiglaConselhoID[pagina*20+19]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1045px; top: 758px;">${guia.STRnumeroConselhoID[pagina*20]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1045px; top: 782px;">${guia.STRnumeroConselhoID[pagina*20+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1045px; top: 807px;">${guia.STRnumeroConselhoID[pagina*20+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1045px; top: 832px;">${guia.STRnumeroConselhoID[pagina*20+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1045px; top: 858px;">${guia.STRnumeroConselhoID[pagina*20+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1045px; top: 882px;">${guia.STRnumeroConselhoID[pagina*20+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1004px; top: 1463px;">${guia.STRnumeroConselhoID[pagina*20+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1004px; top: 1489px;">${guia.STRnumeroConselhoID[pagina*20+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1004px; top: 1515px;">${guia.STRnumeroConselhoID[pagina*20+8]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1004px; top: 1541px;">${guia.STRnumeroConselhoID[pagina*20+9]}</div>
```

${guia.STRnumeroConselhoID[pagina*20+10]}
${guia.STRnumeroConselhoID[pagina*20+11]}
${guia.STRnumeroConselhoID[pagina*20+12]}
${guia.STRnumeroConselhoID[pagina*20+13]}
${guia.STRnumeroConselhoID[pagina*20+14]}
${guia.STRnumeroConselhoID[pagina*20+15]}
${guia.STRnumeroConselhoID[pagina*20+16]}
${guia.STRnumeroConselhoID[pagina*20+17]}
${guia.STRnumeroConselhoID[pagina*20+18]}
${guia.STRnumeroConselhoID[pagina*20+19]}
${guia.STRufConselhoID[pagina*20]}
${guia.STRufConselhoID[pagina*20+1]}
${guia.STRufConselhoID[pagina*20+2]}
${guia.STRufConselhoID[pagina*20+3]}
${guia.STRufConselhoID[pagina*20+4]}
${guia.STRufConselhoID[pagina*20+5]}
${guia.STRufConselhoID[pagina*20+6]}
${guia.STRufConselhoID[pagina*20+7]}
${guia.STRufConselhoID[pagina*20+8]}
${guia.STRufConselhoID[pagina*20+9]}
${guia.STRufConselhoID[pagina*20+10]}
${guia.STRufConselhoID[pagina*20+11]}
${guia.STRufConselhoID[pagina*20+12]}
${guia.STRufConselhoID[pagina*20+13]}
${guia.STRufConselhoID[pagina*20+14]}
${guia.STRufConselhoID[pagina*20+15]}
${guia.STRufConselhoID[pagina*20+16]}
${guia.STRufConselhoID[pagina*20+17]}
${guia.STRufConselhoID[pagina*20+18]}
${guia.STRufConselhoID[pagina*20+19]}

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1286px; top: 758px;">${guia.STRcpfID[pagina*20]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1286px; top: 782px;">${guia.STRcpfID[pagina*20+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1286px; top: 807px;">${guia.STRcpfID[pagina*20+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1286px; top: 832px;">${guia.STRcpfID[pagina*20+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1286px; top: 858px;">${guia.STRcpfID[pagina*20+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1286px; top: 882px;">${guia.STRcpfID[pagina*20+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1463px;">${guia.STRcpfID[pagina*20+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1489px;">${guia.STRcpfID[pagina*20+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1515px;">${guia.STRcpfID[pagina*20+8]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1541px;">${guia.STRcpfID[pagina*20+9]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1567px;">${guia.STRcpfID[pagina*20+10]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1593px;">${guia.STRcpfID[pagina*20+11]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1621px;">${guia.STRcpfID[pagina*20+12]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1647px;">${guia.STRcpfID[pagina*20+13]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1673px;">${guia.STRcpfID[pagina*20+14]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1700px;">${guia.STRcpfID[pagina*20+15]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1726px;">${guia.STRcpfID[pagina*20+16]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1751px;">${guia.STRcpfID[pagina*20+17]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1777px;">${guia.STRcpfID[pagina*20+18]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1248px; top: 1804px;">${guia.STRcpfID[pagina*20+19]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
325px; top: 1867px;">${guia.STRdescricaoOPM[pagina*5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
325px; top: 1894px;">${guia.STRdescricaoOPM[pagina*5+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
325px; top: 1919px;">${guia.STRdescricaoOPM[pagina*5+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
325px; top: 1944px;">${guia.STRdescricaoOPM[pagina*5+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
325px; top: 1972px;">${guia.STRdescricaoOPM[pagina*5+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
908px; top: 1894px;">${guia.STRcodigoBarraOPM[pagina*5+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
908px; top: 1919px;">${guia.STRcodigoBarraOPM[pagina*5+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
908px; top: 1944px;">${guia.STRcodigoBarraOPM[pagina*5+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
908px; top: 1972px;">${guia.STRcodigoBarraOPM[pagina*5+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
47px; top: 929px;">${guia.STRtipoFaturamento}</div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
110px; top: 930px;">${guia.STRtipoFaturamento}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
58px; top: 2058px; line-height: 30pt">${guia.STRobservacao}</div>

<img src="img/guiaResumoInternacao_verso.png" style="display:block; position:absolute; z-
index:5; left: 0px; top: 1100px;">
<!-- verso -->
<script>
</script>

</body>
</html>
```

Nome do arquivo: guiaSolicInternacao.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="js/funcoesGuia.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Guia Solicitação de Internação para impressão</title>

</head>
<body>
<img src="img/guiaSolicitacaoInternacao.png" style="display:block; position:absolute; z-
index:5; left: 0px; top: 0px;">
<div id="template" style="display:block; background-color:transparent; z-index:6; position:
absolute; left: 0px; top: 0px;"></div>

<script>
<!-- campos -->
inserirEmCamposData('${guia.STRdataEmissaoGuia}', 778, 218);
inserirEmCamposNumericos('${guia.STRnumeroCarteira}', 69, 281);
inserirEmCamposData('${guia.STRvalidadeCarteira}', 689, 281);
inserirEmCamposNumericos('${guia.STRnumeroCNS}', 742, 330);
inserirEmCamposNumericos('${guia.STRCNPJC} ${guia.STRcpfC}
${guia.STRcodigoPrestadorNaOperadoraC}', 71, 395);
inserirEmCamposNumericos('${guia.STRCNPJPS} ${guia.STRcpfPS}
${guia.STRcodigoPrestadorNaOperadoraPS}', 72, 508);
inserirEmCamposNumericos('${guia.STRnomeTabela}', 81, 804);
inserirEmCamposNumericos('${guia.STRnomeTabelaSEC}', 203, 804);
inserirEmCamposNumericos('${guia.STRtipoTabelaPES}', 77, 869);
inserirEmCamposNumericos('${guia.STRcodigoPES}', 142, 870);
inserirEmCamposNumericos('guia.STRquantidadeSolicitadaPES', 1003, 868);
inserirEmCamposNumericos('${guia.STRtipoTabelaOPM}', 77, 1043);
inserirEmCamposNumericos('${guia.STRcodigoOPM}', 133, 1043);
inserirEmCamposNumericos('${guia.STRquantidadeSolicitadaOPM}', 693, 1044);
inserirEmCamposNumericos('${guia.STRvalorOPM}', 908, 1044);
inserirEmCamposData('${guia.STRdataProvavelAdmisHosp}', 72, 1221);
inserirEmCamposNumericos('${guia.STRdiasSolicitados}', 412, 602);

</script>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
70px; top: 90px;"><img src="img/logo1.JPG" height="100"></div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
899px; top: 112px;">${guia.STRnumeroGuiaPrestador}
${guia.STRnumeroGuiaOperadora}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
64px; top: 222px;">${guia.STRregistroANS}</div>

<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
523px; top: 284px;">${guia.STRnomePlano}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
63px; top: 334px;">${guia.STRnomeBeneficiario}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
401px; top: 397px;">${guia.STRnomeContratadoC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
971px; top: 399px;">${guia.STRnumeroCNESC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
62px; top: 448px;">${guia.STRnomeProfissionalP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
566px; top: 447px;">${guia.STRsiglaConselhoP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
739px; top: 447px;">${guia.STRnumeroConselhoP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
903px; top: 447px;">${guia.STRufConselhoP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
971px; top: 447px;">${guia.STRcbosP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
396px; top: 510px;">${guia.STRnomePrestadorPS}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
71px; top: 555px;">${guia.STRcaraterInternacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
316px; top: 556px;">${guia.STRtipoInternacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
71px; top: 603px;">${guia.STRregimeInternacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
65px; top: 635px;">${guia.STRindicacaoClinica}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
73px; top: 758px;">${guia.STRtipoDoenca}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
246px; top: 758px;">${guia.STRunidadeTempo}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
515px; top: 757px;">${guia.STRindicadorAcidente}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
391px; top: 869px;">${guia.STRdescricaoPES}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
369px; top: 1042px;">${guia.STRdescricaoOPM}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
754px; top: 1044px;">${guia.STRfabricanteOPM}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
75px; top: 1311px; width: 960px; line-height: 23pt;">${guia.STRobservacao}</div>

</body>
</html>
```

Nome do arquivo: guiaSP_SADT.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="js/funcoesGuia.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Guia SP/SADT para impressão</title>

</head>
<body>
<form method="POST">
<div id="botaopaginacao" style="display:block; background-color:transparent; z-index:7;
position: absolute; left: 50px; top: 0px;">
<c:if test="${pagina>0}">
        <input type="button" value="&lt;&lt;" name="Anterior" onclick="var pag =
document.getElementById('pagina'); pag.value = +pag.value - 1; this.form.submit()">
</c:if>
<c:if test="${pagina<ultimaPagina}">
        <input type="button" value="&gt;&gt;" name="Proxima" onclick="var pag =
document.getElementById('pagina'); pag.value = +pag.value + 1; this.form.submit()">
</c:if>
</div>
<input type="hidden" name="pagina" id="pagina" value="${pagina}">
<c:forEach var="inputVar" items="${parametros}">
<input type="hidden" name="${inputVar.nome}" value="${inputVar.valor}">
</c:forEach>
</form>

<img src="img/guiaSADT.png" style="display:block; position:absolute; z-index:5; left: 0px; top:
0px; page-break-after:always;">
<div id="template" style="display:block; background-color:transparent; z-index:6; position:
absolute; left: 0px; top: 0px;"></div>

<script>
inserirEmCamposData2('${guia.STRdataAutorizacao}',666,135);
inserirEmCamposData2('${guia.STRvalidadeSenha}',962,134);
inserirEmCamposData2('${guia.STRdataEmissaoGuia}',1130,134);
inserirEmCamposData2('${guia.STRvalidadeCarteira}',582,191);
inserirEmCamposData2('${guia.STRdataPR[pagina*5]}',57,743);
inserirEmCamposData2('${guia.STRdataPR[pagina*5+1]}',56,769);
inserirEmCamposData2('${guia.STRdataPR[pagina*5+2]}',57,795);
inserirEmCamposData2('${guia.STRdataPR[pagina*5+3]}',56,822);
inserirEmCamposData2('${guia.STRdataPR[pagina*5+4]}',58,847);

inserirEmCamposNumericos2('${guia.STRnumeroGuiaPrincipal}',240,136);
inserirEmCamposNumericos2('${guia.STRnumeroCarteira}',51,189);
inserirEmCamposNumericos2('${guia.STRnumeroCNS}',1230,191);
inserirEmCamposNumericos2('${guia.STRCNPJC}${guia.STRcpfC}${guia.STRcodigoPrestado
rNaOperadoraC}',52,245);
inserirEmCamposNumericos2('${guia.STRCNPJPE}${guia.STRCPFPE}${guia.STRcodigoPres
tadorNaOperadoraPE}',54,542);
```

```
inserirEmCamposNumericos2('${guia.STRcpfPEC}${guia.STRcodigoPrestadorNaOperadoraP
EC}',54,583);
inserirEmCamposNumericos2('${guia.STRtipoAtendimento}',51,637);
inserirEmCamposNumericos2('${guia.STRindicadorAcidente}',692,638);
inserirEmCamposNumericos2('${guia.STRtipoSaida}',1086,638);
inserirEmCamposNumericos2('${guia.STRtipoDoenca}',50,691);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*5]}',381,743);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*5+1]}',382,769);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*5+2]}',383,794);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*5+3]}',383,822);
inserirEmCamposNumericos2('${guia.STRtipoTabelaPR[pagina*5+4]}',383,848);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*5]}',432,743);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*5+1]}',429,768);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*5+2]}',432,794);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*5+3]}',431,822);
inserirEmCamposNumericos2('${guia.STRcodigoPR[pagina*5+4]}',432,847);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*5]}',934,743);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*5+1]}',934,769);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*5+2]}',934,795);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*5+3]}',934,822);
inserirEmCamposNumericos2('${guia.STRquantidadeRealizadaPR[pagina*5+4]}',937,847);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*5]}',986,743);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*5+1]}',985,770);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*5+2]}',985,795);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*5+3]}',985,821);
inserirEmCamposNumericos2('${guia.STRviaAcessoPR[pagina*5+4]}',985,848);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*5]}',1015,743);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*5+1]}',1017,768);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*5+2]}',1018,795);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*5+3]}',1016,821);
inserirEmCamposNumericos2('${guia.STRtecnicaUtilizadaPR[pagina*5+4]}',1015,848);
inserirEmCamposNumericos2('${guia.STRservicosExecutados}',49,1012);
inserirEmCamposNumericos2('${guia.STRtaxas}',263,1012);
inserirEmCamposNumericos2('${guia.STRmateriais}',468,1012);
inserirEmCamposNumericos2('${guia.STRmedicamentos}',680,1012);
inserirEmCamposNumericos2('${guia.STRdiarias}',892,1012);
inserirEmCamposNumericos2('${guia.STRgases}',1107,1015);
inserirEmCamposNumericos2('${guia.STRtotalGeral}',1319,1014);
inserirEmCamposNumericos2('${guia.STRvalor}',191, 690);

inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*5]}', 192, 743);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*5+1]}', 192, 769);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*5+2]}', 192, 794);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*5+3]}', 192, 822);
inserirEmCamposHora('${guia.STRhoraInicioPR[pagina*5+4]}', 192, 848);

inserirEmCamposHora('${guia.STRhoraFimPR[pagina*5]}', 285, 743);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*5+1]}', 285, 769);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*5+2]}', 285, 794);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*5+3]}', 285, 822);
inserirEmCamposHora('${guia.STRhoraFimPR[pagina*5+4]}', 285, 848);

inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*5]}', 1050, 743);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*5+1]}', 1050, 769);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*5+2]}', 1050, 794);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*5+3]}', 1050, 822);
inserirEmCamposNumericos2('${guia.STRreducaoAcrescimoPR[pagina*5+4]}', 1050, 848);

inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*5]}', 1122, 743);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*5+1]}', 1122, 769);
```

```
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*5+2]}', 1122, 794);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*5+3]}', 1122, 822);
inserirEmCamposNumericos2('${guia.STRvalorPR[pagina*5+4]}', 1122, 848);

inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*5]}', 1269, 743);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*5+1]}', 1269, 769);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*5+2]}', 1269, 794);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*5+3]}', 1269, 822);
inserirEmCamposNumericos2('${guia.STRvalorTotalPR[pagina*5+4]}', 1269, 848);

inserirEmCamposData('${guia.STRdataHoraAtendimento}', 57, 344);
</script>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
60px; top: 50px;"><img src="img/logo1.JPG" height="65" width="120"></div>

<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
48px; top: 136px;">${guia.STRregistroANS}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1334px; top:
71px;">${guia.STRnumeroGuiaPrestador}${guia.STRnumeroGuiaOperadora}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
820px; top: 136px;">${guia.STRsenhaAutorizacao}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
460px; top: 194px;">${guia.STRnomePlano}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
730px; top: 192px;">${guia.STRnomeBeneficiario}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
346px; top: 246px;">${guia.STRnomeContratadoC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
940px; top: 249px;">${guia.STRnumeroCNESC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
43px; top: 291px;">${guia.STRnomeContratadoC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
643px; top: 292px;">${guia.STRsiglaConselhoP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
890px; top: 292px;">${guia.STRnumeroConselhoP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1116px; top: 292px;">${guia.STRufConselhoP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1189px; top: 290px;">${guia.STRcbosP}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
323px; top: 345px;">${guia.STRcaraterAtendimento}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
644px; top: 348px;line-height: 10pt">${guia.STRindicacaoClinica}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
240px; top: 691px;">${guia.STRunidadeTempo}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
344px; top: 545px;">${guia.STRnomeContratadoPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
791px; top: 545px;">${guia.STRtipoLogradouroPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
841px; top: 546px;">${guia.STRlogradouroPE} , ${guia.STRnumeroPE} ,
${guia.STRcomplementoPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1085px; top: 546px;">${guia.STRmunicipioPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1237px; top: 546px;">${guia.STRcodigoUFPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1283px; top: 547px;">${guia.STRcodigoIBGEMunicipioPE}</div>
```

```
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1363px; top: 546px;">${guia.STRcepPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1438px; top: 546px;">${guia.STRnumeroCNESPE}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
345px; top: 586px;">${guia.STRnomeExecutantePEC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
792px; top: 586px;">${guia.STRsiglaConselhoPEC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1014px; top: 588px;">${guia.STRnumeroConselhoPEC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1197px; top: 587px;">${guia.STRufConselhoPEC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
1268px; top: 588px;">${guia.STRcodigoCBOSPEC}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
630px; top: 743px;">${guia.STRdescricaoPR[pagina*5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
633px; top: 769px;">${ guia.STRdescricaoPR[pagina*5+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
632px; top: 795px;">${ guia.STRdescricaoPR[pagina*5+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
632px; top: 822px;">${ guia.STRdescricaoPR[pagina*5+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
631px; top: 847px;">${ guia.STRdescricaoPR[pagina*5+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
48px; top: 942px;line-height: 18pt">${guia.STRobservacao}</div>

<img src="img/guiaSADT_verso.png" style="display:block; position:absolute; z-index:5; left:
0px; top: 1200px;">

<!--verso-->
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1544px;">${guia.STRdescricaoOPM[pagina*9]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1570px;">${guia.STRdescricaoOPM[pagina*9+1]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1596px;">${guia.STRdescricaoOPM[pagina*9+2]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1622px;">${guia.STRdescricaoOPM[pagina*9+3]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1648px;">${guia.STRdescricaoOPM[pagina*9+4]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1674px;">${guia.STRdescricaoOPM[pagina*9+5]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1702px;">${guia.STRdescricaoOPM[pagina*9+6]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1728px;">${guia.STRdescricaoOPM[pagina*9+7]}</div>
<div style="display:block; background-color:transparent; z-index:6; position: absolute; left:
313px; top: 1752px;">${guia.STRdescricaoOPM[pagina*9+8]}</div>
<script>
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9]}', 60, 1544);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+1]}', 60, 1570);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+2]}', 60, 1596);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+3]}', 60, 1622);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+4]}', 60, 1648);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+5]}', 60, 1674);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+6]}', 60, 1702);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+7]}', 60, 1728);
inserirEmCamposNumericos2('${guia.STRtipoTabelaOPM[pagina*9+8]}', 60, 1752);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9]}',111, 1544);
```

```
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+1]}',111, 1570);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+2]}',111, 1597);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+3]}',111, 1623);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+4]}',111, 1650);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+5]}',111, 1677);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+6]}',111, 1704);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+7]}',111, 1731);
inserirEmCamposNumericos2('${guia.STRcodigoOPM[pagina*9+8]}',111, 1758);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9]}', 818, 1544);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+1]}', 818, 1570);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+2]}', 818, 1596);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+3]}', 818, 1622);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+4]}', 818, 1648);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+5]}', 818, 1674);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+6]}', 818, 1702);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+7]}', 818, 1728);
inserirEmCamposNumericos2('${guia.STRquantidadeOPM[pagina*9+8]}', 818, 1752);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9]}', 1126, 1544);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+1]}', 1126, 1570);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+2]}', 1126, 1596);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+3]}', 1126, 1622);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+4]}', 1126, 1648);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+5]}', 1126, 1674);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+6]}', 1126, 1702);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+7]}', 1126, 1728);
inserirEmCamposNumericos2('${guia.STRvalorUnitarioOPM[pagina*9+8]}', 1126, 1752);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9]}', 1298, 1544);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+1]}', 1298, 1570);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+2]}', 1298, 1596);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+3]}', 1298, 1622);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+4]}', 1298, 1648);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+5]}', 1298, 1674);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+6]}', 1298, 1702);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+7]}', 1298, 1728);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPM[pagina*9+8]}', 1298, 1752);
inserirEmCamposNumericos2('${guia.STRvalorTotalOPMOPM}', 1352, 1787);
</script>

</body>
</html>
```

Nome do arquivo: headers.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
        java.util.Map map = request.getParameterMap();
%>
```

```
</body>
</html>
```

Nome do arquivo: menu.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
                <title>TissUFPR</title>
                <link rel="StyleSheet" href="dtree.css" type="text/css" />
                <link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css"
/>
                <script type="text/javascript" src="dtree.js"></script>
        </head>
        <body class="niceform">
                        <img src="img/logo.jpg">
                <div class="dtree">
                        <p><a href="javascript: d.openAll();">Expandir Todos</a> | <a
href="javascript: d.closeAll();">Fechar Todos</a></p>
                        <script type="text/javascript">
                                <!--
                                d = new dTree('d');
                                d.add(0,-1,'Página Inicial','principal.jsp','Voltar a tela
inicial','principal');
                                d.add(1,0,'Cadastros','','','principal');
                                d.add(2,1,'Configurações','SvtConfig.ts','','principal');
                                d.add(3,1,'Usuários','SvtUsuarioPesquisa.ts','','principal');
                                d.add(4,1,'Operadora','SvtOperadoraPesquisa.ts','','principal');
                                d.add(5,1,'Prestador','SvtPrestadorPesquisa.ts','','principal');

        d.add(6,1,'Beneficiário','SvtBeneficiarioPesquisa.ts','','principal');
                                d.add(7,0,'Guias','','','principal');

        d.add(8,7,'Consulta','SvtEscolheOperadora.ts?guia=GuiaConsulta&complex=ct_guiaC
onsulta', '', 'principal');

        d.add(9,7,'SP/SADT','SvtEscolheOperadora.ts?guia=GuiaSPSADT&complex=ct_guiaS
P_SADT', '', 'principal');
                                d.add(10,7,'Solicit. de
Internação','SvtEscolheOperadora.ts?guia=GuiaSolilnternacao&complex=ct_guiaSolicInternac
ao','','principal');
                                d.add(11,7,'Resumo de
Internação','SvtEscolheOperadora.ts?guia=GuiaResumoInternacao&complex=ct_guiaResumoI
nternacao','','principal');
                                d.add(12,7,'Honorário
Individual','SvtEscolheOperadora.ts?guia=GuiaHonorarioIndividual&complex=ct_guiaHonorario
Individual','','principal');
                                d.add(13,7,'Consultar guias','svtConsultarGuia.ts','','principal');
                                document.write(d);
                                //-->
                        </script>
                </div>
        </body>
</html>
```

Nome do arquivo: menuPrincipal.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
        </head>
                <FRAMESET ROWS="8%, 92%" border="1">
                        <FRAME SRC="cabecalho.jsp" NORESIZE SCROLLING="no">
                                <frameset COLS="25%, 75%">
                                <FRAME name="menu" SRC="menu.jsp" NORESIZE
SCROLLING="yes">
                                <FRAME name="principal" SRC="principal.jsp" NORESIZE
SCROLLING="yes">
                                </frameset>
                                <noframes>
                                <body></body>
                        </noframes>
                </FRAMESET>
</html>
```

Nome do arquivo: niceforms.js

```
/*############################################################
Name: Niceforms
Version: 2.0
Author: Lucian Slatineanu
URL: http://www.emblematiq.com/projects/niceforms/

Feel free to use and modify but please keep this copyright intact.
############################################################*/

//Theme Variables - edit these to match your theme
var imagesPath = "img/";
var selectRightWidthSimple = 19;
var selectRightWidthScroll = 2;
var selectMaxHeight = 200;
var textareaTopPadding = 10;
var textareaSidePadding = 10;

//Global Variables
var NF = new Array();
var isIE = false;
var resizeTest = 1;

//Initialization function
function NFInit() {
        try {
                document.execCommand('BackgroundImageCache', false, true);
        } catch(e) {}
        if(!document.getElementById) {return false;}
        //alert("click me first");
        NFDo('start');
}
function NFDo(what) {
```

```
	var niceforms = document.getElementsByTagName('form');
	var identifier = new RegExp('(^| )'+'niceform'+'( |$)');
	if (/MSIE (\d+\.\d+);/.test(navigator.userAgent)) {
		var ieversion=new Number(RegExp.$1);
		if(ieversion < 7) {return false;} //exit script if IE6
		isIE = true;
	}
	for(var q = 0; q < niceforms.length; q++) {if(identifier.test(niceforms[q].className)) {
		if(what == "start") { //Load Niceforms
			NF[q] = new niceform(niceforms[q]);
			niceforms[q].start();
		}
		else { //Unload Niceforms
			niceforms[q].unload();
			NF[q] = "";
		}
	}}
}
function NFFix() {
	NFDo('stop');
	NFDo('start');
}
function niceform(nf) {
	nf._inputText = new Array(); nf._inputRadio = new Array(); nf._inputCheck = new
Array(); nf._inputSubmit = new Array(); nf._inputFile = new Array(); nf._textarea = new Array();
nf._select = new Array(); nf._multiselect = new Array();
	nf.add_inputText = function(obj) {this._inputText[this._inputText.length] = obj;
inputText(obj);}
	nf.add_inputRadio = function(obj) {this._inputRadio[this._inputRadio.length] = obj;
inputRadio(obj);}
	nf.add_inputCheck = function(obj) {this._inputCheck[this._inputCheck.length] = obj;
inputCheck(obj);}
	nf.add_inputSubmit = function(obj) {this._inputSubmit[this._inputSubmit.length] = obj;
inputSubmit(obj);}
	nf.add_inputFile = function(obj) {this._inputFile[this._inputFile.length] = obj;
inputFile(obj);}
	nf.add_textarea = function(obj) {this._textarea[this._textarea.length] = obj;
textarea(obj);}
	nf.add_select = function(obj) {this._select[this._select.length] = obj; selects(obj);}
	nf.add_multiselect = function(obj) {this._multiselect[this._multiselect.length] = obj;
multiSelects(obj);}
	nf.start = function() {
		//Separate and assign elements
		var allInputs = this.getElementsByTagName('input');
		for(var w = 0; w < allInputs.length; w++) {
			switch(allInputs[w].type) {
				case "text": case "password": {this.add_inputText(allInputs[w]);
break;}
				case "radio": {this.add_inputRadio(allInputs[w]); break;}
				case "checkbox": {this.add_inputCheck(allInputs[w]); break;}
				case "submit": case "reset": case "button":
{this.add_inputSubmit(allInputs[w]); break;}
				case "file": {this.add_inputFile(allInputs[w]); break;}
			}
		}
		var allButtons = this.getElementsByTagName('button');
		for(var w = 0; w < allButtons.length; w++) {
			this.add_inputSubmit(allButtons[w]);
		}
		var allTextareas = this.getElementsByTagName('textarea');
```

```
		for(var w = 0; w < allTextareas.length; w++) {
			this.add_textarea(allTextareas[w]);
		}
		var allSelects = this.getElementsByTagName('select');
		for(var w = 0; w < allSelects.length; w++) {
			if(allSelects[w].size == "1") {this.add_select(allSelects[w]);}
			else {this.add_multiselect(allSelects[w]);}
		}
		//Start
		for(w = 0; w < this._inputText.length; w++) {this._inputText[w].init();}
		for(w = 0; w < this._inputRadio.length; w++) {this._inputRadio[w].init();}
		for(w = 0; w < this._inputCheck.length; w++) {this._inputCheck[w].init();}
		for(w = 0; w < this._inputSubmit.length; w++) {this._inputSubmit[w].init();}
		for(w = 0; w < this._inputFile.length; w++) {this._inputFile[w].init();}
		for(w = 0; w < this._textarea.length; w++) {this._textarea[w].init();}
		for(w = 0; w < this._select.length; w++) {this._select[w].init(w);}
		for(w = 0; w < this._multiselect.length; w++) {this._multiselect[w].init(w);}
	}
	nf.unload = function() {
		//Stop
		for(w = 0; w < this._inputText.length; w++) {this._inputText[w].unload();}
		for(w = 0; w < this._inputRadio.length; w++) {this._inputRadio[w].unload();}
		for(w = 0; w < this._inputCheck.length; w++) {this._inputCheck[w].unload();}
		for(w = 0; w < this._inputSubmit.length; w++) {this._inputSubmit[w].unload();}
		for(w = 0; w < this._inputFile.length; w++) {this._inputFile[w].unload();}
		for(w = 0; w < this._textarea.length; w++) {this._textarea[w].unload();}
		for(w = 0; w < this._select.length; w++) {this._select[w].unload();}
		for(w = 0; w < this._multiselect.length; w++) {this._multiselect[w].unload();}
	}
}
function inputText(el) { //extent Text inputs
	el.oldClassName = el.className;
	el.left = document.createElement('img');
	el.left.src = imagesPath + "0.png";
	el.left.className = "NFTextLeft";
	el.right = document.createElement('img');
	el.right.src = imagesPath + "0.png";
	el.right.className = "NFTextRight";
	el.dummy = document.createElement('div');
	el.dummy.className = "NFTextCenter";
	el.onfocus = function() {
		this.dummy.className = "NFTextCenter NFh";
		this.left.className = "NFTextLeft NFh";
		this.right.className = "NFTextRight NFh";
	}
	el.onblur = function() {
		this.dummy.className = "NFTextCenter";
		this.left.className = "NFTextLeft";
		this.right.className = "NFTextRight";
	}
	el.init = function() {
		this.parentNode.insertBefore(this.left, this);
		this.parentNode.insertBefore(this.right, this.nextSibling);
		this.dummy.appendChild(this);
		this.right.parentNode.insertBefore(this.dummy, this.right);
		this.className = "NFText";
	}
	el.unload = function() {
		this.parentNode.parentNode.appendChild(this);
		this.parentNode.removeChild(this.left);
```

```
                this.parentNode.removeChild(this.right);
                this.parentNode.removeChild(this.dummy);
                this.className = this.oldClassName;
        }
}
function inputRadio(el) { //extent Radio buttons
        el.oldClassName = el.className;
        el.dummy = document.createElement('div');
        if(el.checked) {el.dummy.className = "NFRadio NFh";}
        else {el.dummy.className = "NFRadio";}
        el.dummy.ref = el;
        if(isIE == false) {el.dummy.style.left = findPosX(el) + 'px'; el.dummy.style.top =
findPosY(el) + 'px';}
        else {el.dummy.style.left = findPosX(el) + 4 + 'px'; el.dummy.style.top = findPosY(el) +
4 + 'px';}
        el.dummy.onclick = function() {
                if(!this.ref.checked) {
                        var siblings = getInputsByName(this.ref.name);
                        for(var q = 0; q < siblings.length; q++) {
                                siblings[q].checked = false;
                                siblings[q].dummy.className = "NFRadio";
                        }
                        this.ref.checked = true;
                        this.className = "NFRadio NFh";
                }
        }
        el.onclick = function() {
                if(this.checked) {
                        var siblings = getInputsByName(this.name);
                        for(var q = 0; q < siblings.length; q++) {
                                siblings[q].dummy.className = "NFRadio";
                        }
                        this.dummy.className = "NFRadio NFh";
                }
        }
        el.onfocus = function() {this.dummy.className += " NFfocused";}
        el.onblur = function() {this.dummy.className = this.dummy.className.replace(/
NFfocused/g, "");}
        el.init = function() {
                this.parentNode.insertBefore(this.dummy, this);
                el.className = "NFhidden";
        }
        el.unload = function() {
                this.parentNode.removeChild(this.dummy);
                this.className = this.oldClassName;
        }
}
function inputCheck(el) { //extend Checkboxes
        el.oldClassName = el.className;
        el.dummy = document.createElement('img');
        el.dummy.src = imagesPath + "0.png";
        if(el.checked) {el.dummy.className = "NFCheck NFh";}
        else {el.dummy.className = "NFCheck";}
        el.dummy.ref = el;
        if(isIE == false) {el.dummy.style.left = findPosX(el) + 'px'; el.dummy.style.top =
findPosY(el) + 'px';}
        else {el.dummy.style.left = findPosX(el) + 4 + 'px'; el.dummy.style.top = findPosY(el) +
4 + 'px';}
        el.dummy.onclick = function() {
                if(!this.ref.checked) {
```

```
                        this.ref.checked = true;
                        this.className = "NFCheck NFh";
                }
                else {
                        this.ref.checked = false;
                        this.className = "NFCheck";
                }
        }
        el.onclick = function() {
                if(this.checked) {this.dummy.className = "NFCheck NFh";}
                else {this.dummy.className = "NFCheck";}
        }
        el.onfocus = function() {this.dummy.className += " NFfocused";}
        el.onblur = function() {this.dummy.className = this.dummy.className.replace(/
NFfocused/g, "");}
        el.init = function() {
                this.parentNode.insertBefore(this.dummy, this);
                el.className = "NFhidden";
        }
        el.unload = function() {
                this.parentNode.removeChild(this.dummy);
                this.className = this.oldClassName;
        }
}
function inputSubmit(el) { //extend Buttons
        el.oldClassName = el.className;
        el.left = document.createElement('img');
        el.left.className = "NFButtonLeft";
        el.left.src = imagesPath + "0.png";
        el.right = document.createElement('img');
        el.right.src = imagesPath + "0.png";
        el.right.className = "NFButtonRight";
        el.onmouseover = function() {
                this.className = "NFButton NFh";
                this.left.className = "NFButtonLeft NFh";
                this.right.className = "NFButtonRight NFh";
        }
        el.onmouseout = function() {
                this.className = "NFButton";
                this.left.className = "NFButtonLeft";
                this.right.className = "NFButtonRight";
        }
        el.init = function() {
                this.parentNode.insertBefore(this.left, this);
                this.parentNode.insertBefore(this.right, this.nextSibling);
                this.className = "NFButton";
        }
        el.unload = function() {
                this.parentNode.removeChild(this.left);
                this.parentNode.removeChild(this.right);
                this.className = this.oldClassName;
        }
}
function inputFile(el) { //extend File inputs
        el.oldClassName = el.className;
        el.dummy = document.createElement('div');
        el.dummy.className = "NFFile";
        el.file = document.createElement('div');
        el.file.className = "NFFileNew";
        el.center = document.createElement('div');
```

```
		el.center.className = "NFTextCenter";
		el.clone = document.createElement('input');
		el.clone.type = "text";
		el.clone.className = "NFText";
		el.clone.ref = el;
		el.left = document.createElement('img');
		el.left.src = imagesPath + "0.png";
		el.left.className = "NFTextLeft";
		el.button = document.createElement('img');
		el.button.src = imagesPath + "0.png";
		el.button.className = "NFFileButton";
		el.button.ref = el;
		el.button.onclick = function() {this.ref.click();}
		el.init = function() {
			var top = this.parentNode;
			if(this.previousSibling) {var where = this.previousSibling;}
			else {var where = top.childNodes[0];}
			top.insertBefore(this.dummy, where);
			this.dummy.appendChild(this);
			this.center.appendChild(this.clone);
			this.file.appendChild(this.center);
			this.file.insertBefore(this.left, this.center);
			this.file.appendChild(this.button);
			this.dummy.appendChild(this.file);
			this.className = "NFhidden";
			this.relatedElement = this.clone;
		}
		el.unload = function() {
			this.parentNode.parentNode.appendChild(this);
			this.parentNode.removeChild(this.dummy);
			this.className = this.oldClassName;
		}
		el.onchange = el.onmouseout = function() {this.relatedElement.value = this.value;}
		el.onfocus = function() {
			this.left.className = "NFTextLeft NFh";
			this.center.className = "NFTextCenter NFh";
			this.button.className = "NFFileButton NFh";
		}
		el.onblur = function() {
			this.left.className = "NFTextLeft";
			this.center.className = "NFTextCenter";
			this.button.className = "NFFileButton";
		}
		el.onselect = function() {
			this.relatedElement.select();
			this.value = '';
		}
}
function textarea(el) { //extend Textareas
		el.oldClassName = el.className;
		el.height = el.offsetHeight - textareaTopPadding;
		el.width = el.offsetWidth - textareaSidePadding;
		el.topLeft = document.createElement('img');
		el.topLeft.src = imagesPath + "0.png";
		el.topLeft.className = "NFTextareaTopLeft";
		el.topRight = document.createElement('div');
		el.topRight.className = "NFTextareaTop";
		el.bottomLeft = document.createElement('img');
		el.bottomLeft.src = imagesPath + "0.png";
		el.bottomLeft.className = "NFTextareaBottomLeft";
```

```
		el.bottomRight = document.createElement('div');
		el.bottomRight.className = "NFTextareaBottom";
		el.left = document.createElement('div');
		el.left.className = "NFTextareaLeft";
		el.right = document.createElement('div');
		el.right.className = "NFTextareaRight";
		el.init = function() {
			var top = this.parentNode;
			if(this.previousSibling) {var where = this.previousSibling;}
			else {var where = top.childNodes[0];}
			top.insertBefore(el.topRight, where);
			top.insertBefore(el.right, where);
			top.insertBefore(el.bottomRight, where);
			this.topRight.appendChild(this.topLeft);
			this.right.appendChild(this.left);
			this.right.appendChild(this);
			this.bottomRight.appendChild(this.bottomLeft);
			el.style.width = el.topRight.style.width = el.bottomRight.style.width = el.width +
'px';
			el.style.height = el.left.style.height = el.right.style.height = el.height + 'px';
			this.className = "NFTextarea";
		}
		el.unload = function() {
			this.parentNode.parentNode.appendChild(this);
			this.parentNode.removeChild(this.topRight);
			this.parentNode.removeChild(this.bottomRight);
			this.parentNode.removeChild(this.right);
			this.className = this.oldClassName;
			this.style.width = this.style.height = "";
		}
		el.onfocus = function() {
			this.topLeft.className = "NFTextareaTopLeft NFh";
			this.topRight.className = "NFTextareaTop NFhr";
			this.left.className = "NFTextareaLeftH";
			this.right.className = "NFTextareaRightH";
			this.bottomLeft.className = "NFTextareaBottomLeft NFh";
			this.bottomRight.className = "NFTextareaBottom NFhr";
		}
		el.onblur = function() {
			this.topLeft.className = "NFTextareaTopLeft";
			this.topRight.className = "NFTextareaTop";
			this.left.className = "NFTextareaLeft";
			this.right.className = "NFTextareaRight";
			this.bottomLeft.className = "NFTextareaBottomLeft";
			this.bottomRight.className = "NFTextareaBottom";
		}
}
function selects(el) { //extend Selects
		el.oldClassName = el.className;
		el.dummy = document.createElement('div');
		el.dummy.className = "NFSelect";
		el.dummy.style.width = el.offsetWidth + 'px';
		el.dummy.ref = el;
		el.left = document.createElement('img');
		el.left.src = imagesPath + "0.png";
		el.left.className = "NFSelectLeft";
		el.right = document.createElement('div');
		el.right.className = "NFSelectRight";
		el.txt = document.createTextNode(el.options[0].text);
		el.bg = document.createElement('div');
```

```
		el.bg.className = "NFSelectTarget";
		el.bg.style.display = "none";
		el.opt = document.createElement('ul');
		el.opt.className = "NFSelectOptions";
		el.dummy.style.left = findPosX(el) + 'px';
		el.dummy.style.top = findPosY(el) + 'px';
		el.opts = new Array(el.options.length);
		el.init = function(pos) {
			this.dummy.appendChild(this.left);
			this.right.appendChild(this.txt);
			this.dummy.appendChild(this.right);
			this.bg.appendChild(this.opt);
			this.dummy.appendChild(this.bg);
			for(var q = 0; q < this.options.length; q++) {
				this.opts[q] = new option(this.options[q], q);
				this.opt.appendChild(this.options[q].li);
				this.options[q].lnk.onclick = function() {
					this._onclick();
					this.ref.dummy.getElementsByTagName('div')[0].innerHTML =
this.ref.options[this.pos].text;
					this.ref.options[this.pos].selected = "selected";
					for(var w = 0; w < this.ref.options.length; w++)
{this.ref.options[w].lnk.className = "";}
					this.ref.options[this.pos].lnk.className = "NFOptionActive";
				}
			}
			if(this.options.selectedIndex) {
				this.dummy.getElementsByTagName('div')[0].innerHTML =
this.options[this.options.selectedIndex].text;
				this.options[this.options.selectedIndex].lnk.className =
"NFOptionActive";
			}
			this.dummy.style.zIndex = 999 - pos;
			this.parentNode.insertBefore(this.dummy, this);
			this.className = "NFhidden";
		}
		el.unload = function() {
			this.parentNode.removeChild(this.dummy);
			this.className = this.oldClassName;
		}
		el.dummy.onclick = function() {
			var allDivs = document.getElementsByTagName('div'); for(var q = 0; q <
allDivs.length; q++) {if((allDivs[q].className == "NFSelectTarget") && (allDivs[q] != this.ref.bg))
{allDivs[q].style.display = "none";}}
			if(this.ref.bg.style.display == "none") {this.ref.bg.style.display = "block";}
			else {this.ref.bg.style.display = "none";}
			if(this.ref.opt.offsetHeight > selectMaxHeight) {
				this.ref.bg.style.width = this.ref.offsetWidth - selectRightWidthScroll +
33 + 'px';
				this.ref.opt.style.width = this.ref.offsetWidth - selectRightWidthScroll +
'px';
			}
			else {
				this.ref.bg.style.width = this.ref.offsetWidth - selectRightWidthSimple +
33 + 'px';
				this.ref.opt.style.width = this.ref.offsetWidth - selectRightWidthSimple +
'px';
			}
		}
		el.bg.onmouseout = function(e) {
```

```
                if (!e) var e = window.event;
                e.cancelBubble = true;
                if (e.stopPropagation) e.stopPropagation();
                var reltg = (e.relatedTarget) ? e.relatedTarget : e.toElement;
                if((reltg.nodeName == 'A') || (reltg.nodeName == 'LI') || (reltg.nodeName ==
'UL')) return;
                if((reltg.nodeName == 'DIV') || (reltg.className == 'NFSelectTarget')) return;
                else{this.style.display = "none";}
        }
        el.dummy.onmouseout = function(e) {
                if (!e) var e = window.event;
                e.cancelBubble = true;
                if (e.stopPropagation) e.stopPropagation();
                var reltg = (e.relatedTarget) ? e.relatedTarget : e.toElement;
                if((reltg.nodeName == 'A') || (reltg.nodeName == 'LI') || (reltg.nodeName ==
'UL')) return;
                if((reltg.nodeName == 'DIV') || (reltg.className == 'NFSelectTarget')) return;
                else{this.ref.bg.style.display = "none";}
        }
        el.onfocus = function() {this.dummy.className += " NFfocused";}
        el.onblur = function() {this.dummy.className = this.dummy.className.replace(/
NFfocused/g, "");}
        el.onkeydown = function(e) {
                if (!e) var e = window.event;
                var thecode = e.keyCode;
                var active = this.selectedIndex;
                switch(thecode){
                        case 40: //down
                                if(active < this.options.length - 1) {
                                        for(var w = 0; w < this.options.length; w++)
{this.options[w].lnk.className = "";}
                                        var newOne = active + 1;
                                        this.options[newOne].selected = "selected";
                                        this.options[newOne].lnk.className =
"NFOptionActive";

        this.dummy.getElementsByTagName('div')[0].innerHTML = this.options[newOne].text;
                                }
                                return false;
                                break;
                        case 38: //up
                                if(active > 0) {
                                        for(var w = 0; w < this.options.length; w++)
{this.options[w].lnk.className = "";}
                                        var newOne = active - 1;
                                        this.options[newOne].selected = "selected";
                                        this.options[newOne].lnk.className =
"NFOptionActive";

        this.dummy.getElementsByTagName('div')[0].innerHTML = this.options[newOne].text;
                                }
                                return false;
                                break;
                        default:
                                break;
                }
        }
}
function multiSelects(el) { //extend Multiple Selects
        el.oldClassName = el.className;
```

```
		el.height = el.offsetHeight;
		el.width = el.offsetWidth;
		el.topLeft = document.createElement('img');
		el.topLeft.src = imagesPath + "0.png";
		el.topLeft.className = "NFMultiSelectTopLeft";
		el.topRight = document.createElement('div');
		el.topRight.className = "NFMultiSelectTop";
		el.bottomLeft = document.createElement('img');
		el.bottomLeft.src = imagesPath + "0.png";
		el.bottomLeft.className = "NFMultiSelectBottomLeft";
		el.bottomRight = document.createElement('div');
		el.bottomRight.className = "NFMultiSelectBottom";
		el.left = document.createElement('div');
		el.left.className = "NFMultiSelectLeft";
		el.right = document.createElement('div');
		el.right.className = "NFMultiSelectRight";
		el.init = function() {
			var top = this.parentNode;
			if(this.previousSibling) {var where = this.previousSibling;}
			else {var where = top.childNodes[0];}
			top.insertBefore(el.topRight, where);
			top.insertBefore(el.right, where);
			top.insertBefore(el.bottomRight, where);
			this.topRight.appendChild(this.topLeft);
			this.right.appendChild(this.left);
			this.right.appendChild(this);
			this.bottomRight.appendChild(this.bottomLeft);
			el.style.width = el.topRight.style.width = el.bottomRight.style.width = el.width +
'px';
			el.style.height = el.left.style.height = el.right.style.height = el.height + 'px';
			el.className = "NFMultiSelect";
		}
		el.unload = function() {
			this.parentNode.parentNode.appendChild(this);
			this.parentNode.removeChild(this.topRight);
			this.parentNode.removeChild(this.bottomRight);
			this.parentNode.removeChild(this.right);
			this.className = this.oldClassName;
			this.style.width = this.style.height = "";
		}
		el.onfocus = function() {
			this.topLeft.className = "NFMultiSelectTopLeft NFh";
			this.topRight.className = "NFMultiSelectTop NFhr";
			this.left.className = "NFMultiSelectLeftH";
			this.right.className = "NFMultiSelectRightH";
			this.bottomLeft.className = "NFMultiSelectBottomLeft NFh";
			this.bottomRight.className = "NFMultiSelectBottom NFhr";
		}
		el.onblur = function() {
			this.topLeft.className = "NFMultiSelectTopLeft";
			this.topRight.className = "NFMultiSelectTop";
			this.left.className = "NFMultiSelectLeft";
			this.right.className = "NFMultiSelectRight";
			this.bottomLeft.className = "NFMultiSelectBottomLeft";
			this.bottomRight.className = "NFMultiSelectBottom";
		}
}
function option(el, no) { //extend Options
		el.li = document.createElement('li');
		el.lnk = document.createElement('a');
```

```
        el.lnk.href = "javascript:;";
        el.lnk.ref = el.parentNode;
        el.lnk.pos = no;
        el.lnk._onclick = el.onclick || function () {};
        el.txt = document.createTextNode(el.text);
        el.lnk.appendChild(el.txt);
        el.li.appendChild(el.lnk);
}

//Get Position
function findPosY(obj) {
        var posTop = 0;
        do {posTop += obj.offsetTop;} while (obj = obj.offsetParent);
        return posTop;
}
function findPosX(obj) {
        var posLeft = 0;
        do {posLeft += obj.offsetLeft;} while (obj = obj.offsetParent);
        return posLeft;
}
//Get Siblings
function getInputsByName(name) {
        var inputs = document.getElementsByTagName("input");
        var w = 0; var results = new Array();
        for(var q = 0; q < inputs.length; q++) {if(inputs[q].name == name) {results[w] = inputs[q];
++w;}}
        return results;
}

//Add events
var existingLoadEvent = window.onload || function () {};
var existingResizeEvent = window.onresize || function() {};
window.onload = function () {
  existingLoadEvent();
  NFInit();
}
window.onresize = function() {
        if(resizeTest != document.documentElement.clientHeight) {
                existingResizeEvent();
                NFFix();
        }
        resizeTest = document.documentElement.clientHeight;
}
```

Nome do arquivo: niceforms-default.css

```
/*Defaults Styling*/
body {font:12px/17px Arial, Helvetica, sans-serif; color:#333; background:#ccc; padding:40px
5px 5px 5px;}
fieldset {background:#f2f2e6; padding:10px; border:1px solid #fff; border-color:#fff #666661
#666661 #fff; margin-bottom:36px; width:700px;}
input, textarea, select {font:12px/12px Arial, Helvetica, sans-serif; padding:0;}
fieldset.action {background:#9da2a6; border-color:#e5e5e5 #797c80 #797c80 #e5e5e5;
margin-top:-20px;}
legend {background:#bfbf30; color:#fff; font:17px/21px Calibri, Arial, Helvetica, sans-serif;
padding:0 10px; margin:-26px 0 0 -11px; font-weight:bold; border:1px solid #fff; border-
color:#e5e5c3 #505014 #505014 #e5e5c3;}
label {font-size:11px; font-weight:bold; color:#666;}
label.opt {font-weight:normal;}
```

```
dl {clear:both;}
dt {float:left; text-align:right; width:90px; line-height:25px; margin:0 10px 10px 0;}
dd {float:left; width:475px; line-height:25px; margin:0 0 10px 0;}
#footer {font-size:11px;}

#container {width:700px; margin:0 auto;}

/*##########################################
Script: Niceforms 2.0
Theme: StandardBlue
Author: Lucian Slatineanu
URL: http://www.emblematiq.com/
##########################################*/

/*Text inputs*/
.NFText {border:none; vertical-align:middle; font:12px/15px Arial, Helvetica, sans-serif;
background:none;}
.NFTextCenter {height:15px; background:url(img/input.png) repeat-x 0 0; padding:3px 0;
margin:0; float:left; line-height:15px;}
.NFTextLeft, .NFTextRight {width:7px; height:21px; vertical-align:middle; float:left;}
.NFTextLeft {background:url(img/input-left.png) no-repeat 0 0;}
.NFTextRight {background:url(img/input-right.png) no-repeat 0 0;}
/*Radio*/
.NFRadio {cursor:pointer; position:absolute; display:block; width:13px; height:13px; border:1px
solid transparent; background:url(img/radio.png) no-repeat 0 0; z-index:2;}
/*Checkbox*/
.NFCheck {cursor:pointer; position:absolute; width:12px; height:12px; border:1px solid
transparent; background:url(img/checkbox.png) no-repeat 0 0; z-index:2;}
/*Buttons*/
.NFButton {width:auto; height:26px; color:#fff; padding:0 2px; background:url(img/button.png)
repeat-x 0 0; cursor:pointer; border:none; font:10px/26px Tahoma, Arial, Helvetica, sans-serif;
font-weight:bold; text-transform:uppercase; letter-spacing:1px; vertical-align:middle;}
.NFButtonLeft, .NFButtonRight {width:6px; height:26px; vertical-align:middle;}
.NFButtonLeft {background:url(img/button-left.png) no-repeat 0 0;}
.NFButtonRight {background:url(img/button-right.png) no-repeat 0 0;}
/*Textareas*/
.NFTextarea {border:none; background:none; font:12px/12px Arial, Helvetica, sans-serif;
margin:0;}
.NFTextareaTop, .NFTextareaBottom {height:5px; clear:both; float:none; padding-right:10px;}
.NFTextareaTop {background:url(img/textarea-tr.png) no-repeat 100% 0;}
.NFTextareaBottom {background:url(img/textarea-br.png) no-repeat 100% 0; margin-
bottom:5px;}
.NFTextareaTopLeft, .NFTextareaBottomLeft {width:5px; height:5px;}
.NFTextareaTopLeft {background:#f2f2e6 url(img/textarea-tl.png) no-repeat 0 0;}
.NFTextareaBottomLeft {background:#f2f2e6 url(img/textarea-bl.png) no-repeat 0 0;}
.NFTextareaLeft, .NFTextareaRight, .NFTextareaLeftH, .NFTextareaRightH {float:left; padding-
bottom:5px;}
.NFTextareaLeft, .NFTextareaLeftH {width:5px;}
.NFTextareaLeft {background:url(img/textarea-l-off.png) repeat-y 0 0;}
.NFTextareaLeftH {background:url(img/textarea-l-over.png) repeat-y 0 0;}
.NFTextareaRight, .NFTextareaRightH {padding-right:5px; padding-bottom:0;}
.NFTextareaRight {background:url(img/textarea-r-off.png) repeat-y 100% 0;}
.NFTextareaRightH {background:url(img/textarea-r-over.png) repeat-y 100% 100%;}
/*Files*/
.NFFileButton {padding-bottom:0; vertical-align:bottom; cursor:pointer;
background:url(img/file.png) no-repeat 0 0; width:60px; height:21px;}
.NFFile {position:relative; margin-bottom:5px;}
.NFFile input.NFhidden {position:relative; filter:alpha(opacity=0); opacity:0; z-index:2;
cursor:pointer; text-align:left;}
.NFFileNew {position:absolute; top:0px; left:0px; z-index:1;}
```

```
/*Selects*/
.NFSelect {height:21px; position:absolute; border:1px solid transparent;}
.NFSelectLeft {float:left; width:3px; height:21px; background:url(img/select-left.png) no-repeat 0
0; vertical-align:middle;}
.NFSelectRight {height:21px; width:auto; background:url(img/select-right.png) no-repeat 100%
0; cursor:pointer; font:12px/21px Arial, Helvetica, sans-serif; color:#fff; padding-left:3px;
margin-left:3px;}
.NFSelectTarget {position:absolute; background:none; margin-left:-13px; margin-top:18px; z-
index:3; left:0; top:0; padding-bottom:13px;}
.NFSelectOptions {position:relative; background:#707175; margin-left:16px; margin-top:0; list-
style:none; padding:4px 0; color:#fff; font:11px/13px Arial, Helvetica, sans-serif; z-index:4; max-
height:200px; overflow-y:auto; overflow-x:hidden; left:0; top:0;}
.NFSelectOptions li {padding-bottom:1px;}
.NFSelectOptions a {display:block; text-decoration:none; color:#fff; padding:2px 3px;
background:none;}
.NFSelectOptions a.NFOptionActive {background:#464646;}
.NFSelectOptions a:hover {background:#333;}
/*Multiple Selects*/
.NFMultiSelect {border:0; background:none; margin:0;}
.NFMultiSelectTop, .NFMultiSelectBottom {height:5px; clear:both; float:none; padding-
right:10px;}
.NFMultiSelectTop {background:url(img/textarea-tr.png) no-repeat 100% 0;}
.NFMultiSelectBottom {background:url(img/textarea-br.png) no-repeat 100% 0; margin-
bottom:5px;}
.NFMultiSelectTopLeft, .NFMultiSelectBottomLeft {width:5px; height:5px;}
.NFMultiSelectTopLeft {background:#f2f2e6 url(img/textarea-tl.png) no-repeat 0 0;}
.NFMultiSelectBottomLeft {background:#f2f2e6 url(img/textarea-bl.png) no-repeat 0 0;}
.NFMultiSelectLeft, .NFMultiSelectRight, .NFMultiSelectLeftH, .NFMultiSelectRightH {float:left;
padding-bottom:5px;}
.NFMultiSelectLeft, .NFMultiSelectLeftH {width:5px;}
.NFMultiSelectLeft {background:url(img/textarea-l-off.png) repeat-y 0 0;}
.NFMultiSelectLeftH {background:url(img/textarea-l-over.png) repeat-y 0 0;}
.NFMultiSelectRight, .NFMultiSelectRightH {padding-right:5px; padding-bottom:0;}
.NFMultiSelectRight {background:url(img/textarea-r-off.png) repeat-y 100% 0;}
.NFMultiSelectRightH {background:url(img/textarea-r-over.png) repeat-y 100% 0;}

/*Focused*/
.NFfocused {border:1px dotted #666;}
/*Hovered*/
.NFh {background-position:0 100%;}
.NFhr {background-position:100% 100%;}
/*Hidden*/
.NFhidden {opacity:0; z-index:-1; position:relative;}
/*Safari*/
select, input, textarea, button {outline:none; resize:none;}
```

Nome do arquivo: pesquisaBen.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pesquisa :: Beneficiario</title>
<script language="JavaScript" src="js/funcoesBeneficiario.js"> </script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
```

```
</head>
<body>
	<c:if test="${!empty alerta}">
		<script language="JavaScript">
			alert('${alerta}');
		</script>
	</c:if>
	<form id="formEditar" name="formEditar" method="post"
action="SvtBeneficiarioCadastro.ts">
		<input type=hidden name="operacao" value="editar">
		<input type=hidden name="id" value="">
	</form>
	<form id="formExcluir" name="formExcluir" method="post"
action="SvtBeneficiarioExcluir.ts">
		<input type=hidden name="id" value="">
	</form>
	<form id="filtro" name="filtro" method="post" action="SvtBeneficiarioPesquisa.ts"
class="niceform">
	<h3>Pesquisa de Beneficiários</h3>
	<fieldset><legend>Filtro de Pesquisa</legend>
	<div class="linha1"><label>Nome: <input name="nome"
	type="text" id="nome" size="70" maxlength="70"></label>
	</div>
	<input type="submit" name="pesquisar" value="Pesquisar"></fieldset>
	</form>
	<c:if test="${!empty listaBeneficiarios}">
		<table border="1">
			<tr>
				<th>Nome</th>
				<th>Número da Carteira</th>
				<th>Validade da Carteira</th>
				<th>Tipo de Plano</th>
				<th>Número Cartão Nacional</th>
				<th>Alterar</th>
				<th>Excluir</th>
			</tr>
			<c:forEach var="beneficiario" items="${listaBeneficiarios}">
				<tr>
					<td>${beneficiario.nome}</td>
					<td>${beneficiario.numcarteira}</td>
					<td>${beneficiario.valcarteira}</td>
					<td>${beneficiario.tpplano}</td>
					<td>${beneficiario.numcartaonac}</td>
					<td><a
href="javascript:editarBeneficiario(${beneficiario.codigo})"><img
src="img/alterar.png" border=0></a></td>
		<td><a href="javascript:removerBeneficiario(${beneficiario.codigo})"><img
src="img/stop.gif" border=0></a></td>
				</tr>
			</c:forEach>
		</table>
	</c:if>
	<form id="form2" name="form2" method="post" action="SvtBeneficiarioCadastro.ts">
		<input type="Submit" name="CadastrarNovoBeneficiario" value="Novo
Cadastro">
	</form>
</body>
</html>
```

Nome do arquivo: PesquisaEnder.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pesquisa :: Usuários</title>
<script language= "JavaScript" src="js/funcoesEndereco.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <c:if test="${!empty alerta}">
                <script language="JavaScript">
                        alert('${alerta}');
                </script>
        </c:if>
        <form id="formEditar" name="formEditar" method="post" action="SvtEndereco.ts">
                <input type=hidden name="operacao" value="editar">
                <input type=hidden name="cpfcnpj" value="">
                <input type=hidden name="codigo" value="">
        </form>
        <form id="formExcluir" name="formExcluir" method="post" action="SvtEndereco.ts">
                <input type=hidden name="operacao" value="excluir">
                <input type=hidden name="cpfcnpj" value="">
                <input type=hidden name="codigo" value="">
        </form>
        <form id="filtro" name="filtro" method="post" action="ServletEnderecoPesq.ts"
class="niceform">
        <h3> Pesquisa de Endereços </h3>
        <fieldset><legend>Filtro de Pesquisa</legend>
        <input type=hidden name="cpfcnpj" value="${cpfcnpj}">
        <div class="linha">
                <label>Logradouro
                        <input name="log" type="text" id="log" size="70" maxlength="70"
value="${log}">
                </label>
        </div>
        <input type="Submit" name="pesquisar" value="Pesquisar" /></fieldset>
        </form>
        <c:if test="${!empty listaEnderecos}">
                <table border="1">
                        <tr>
                                <th>Logradouro</th>
                                <th>CEP</th>
                                <th>Município</th>
                                <th>Principal</th>
                                <th>Alterar</th>
                                <th>Excluir</th>
                        </tr>
                        <c:forEach var="endereco" items="${listaEnderecos}">
                                <tr>
                                        <td>${endereco.logradouro}</td>
                                        <td>${endereco.cep}</td>
                                        <td>${endereco.municipio}</td>
                                        <td>${endereco.principal}</td>
```

```
                                        <td><a
href="javascript:editarEndereco('${endereco.CFPCNPJ}', ${endereco.codigo})"><img
                                                src="img/alterar.png" border=0></a></td>
            <td><a href="javascript:removerEndereco('${endereco.CFPCNPJ}',
${endereco.codigo})"><img
                                                src="img/stop.gif" border=0></a></td>
                                    </tr>
                                </c:forEach>
                        </table>
                </c:if>
                <form id="form2" name="form2" method="post"
action="SvtEndereco.ts?cpfcnpj=${cpfcnpj}&codigo=0">
                        <input type="Submit" name="cadastrarnovo" value="Novo Cadastro">
                        <input type="Button" name="cancelar" value="Cancelar" onclick="history.go(-
1)">
                </form>
</body>
</html>
```

Nome do arquivo: pesquisaOpe.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pesquisa :: Operadora</title>
<script language="JavaScript" src="js/funcoesOperadora.js"> </script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <c:if test="${!empty alerta}">
                <script language="JavaScript">
                        alert('${alerta}');
                </script>
        </c:if>
        <form id="formEditar" name="formEditar" method="post"
action="SvtOperadoraCadastro.ts">
                <input type=hidden name="operacao" value="editar">
                <input type=hidden name="id" value="">
        </form>
        <form id="formExcluir" name="formExcluir" method="post"
action="SvtOperadoraExcluir.ts">
                <input type=hidden name="id" value="">
        </form>
        <form id="filtro" name="filtro" method="post" action="SvtOperadoraPesquisa.ts"
class="niceform">
        <h3> Pesquisa de Operadoras </h3>
        <fieldset><legend>Filtro de Pesquisa</legend>
        <div class="linha"><label>Nome: <input name="nome"
                type="text" id="nome" size="70" maxlength="70" value="${nome}"> </label>
        </div>
        <input type="Submit" name="pesquisar" value="Pesquisar" /></fieldset>
        </form>
        <c:if test="${!empty listaOperadoras}">
                <table border="1">
```

```
                        <tr>
                                <th>Registro</th>
                                <th>Nome Operadora</th>
                                <th>Alterar</th>
                                <th>Excluir</th>
                        </tr>
                        <c:forEach var="operadora" items="${listaOperadoras}">
                                <tr>
                                        <td>${operadora.registro}</td>
                                        <td>${operadora.nome}</td>
                                        <td><a
href="javascript:editarOperadora(${operadora.codigo})"><img
                                                src="img/alterar.png" border=0></a></td>
                <td><a href="javascript:removerOperadora(${operadora.codigo})"><img
                                                src="img/stop.gif" border=0></a></td>
                                </tr>
                        </c:forEach>
                </table>
        </c:if>
        <form id="form2" name="form2" method="post" action="SvtOperadoraCadastro.ts">
                <input type="Submit" name="CadastrarNovaOperadora" value="Novo
Cadastro">
        </form>
</body>
</html>
```

Nome do arquivo: pesquisaPre.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pesquisa :: Prestador</title>
<script language="JavaScript" src="js/funcoesPrestador.js"> </script>
<link rel="StyleSheet" href="prestador.css" type="text/css" />
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <c:if test="${!empty alerta}">
                <script language="JavaScript">
                        alert('${alerta}');
                </script>
        </c:if>
        <form id="formEditarFisico" name="formEditarFisico" method="post"
action="SvtPrestadorFisicoCadastro.ts">
        <h3> Pesquisa de Prestadores </h3>
                <input type=hidden name="operacao" value="editar">
                <input type=hidden name="cpf" value="">
        </form>
        <form id="formEditarJuridico" name="formEditarJuridico" method="post"
action="SvtPrestadorJuridicoCadastro.ts">
                <input type=hidden name="operacao" value="editar">
                <input type=hidden name="cnpj" value="">
        </form>
```

```
        <form id="formExcluir" name="formExcluir" method="post"
action="SvtPrestadorExcluir.ts">
                <input type=hidden name="id" value="">
        </form>

        <form id="filtro" name="filtro" method="post" action="SvtPrestadorPesquisa.ts"
class="niceform">
        <fieldset><legend>Filtro de Pesquisa</legend>
        <div class="linha1"><label>Nome: <input name="nome"
        type="text" id="nome" size="70" maxlength="70" value="${nome}"></label>
        <label>Tipo:<select name="opcao">
                                <option value="%">Todos</option>
                                <option value="fisico">Fisico</option>
                                <option value="juridico">Juridico</option>
                               </select>
        </label>
        </div>
        <input type="submit" name="pesquisar" value="Pesquisar"></fieldset>
        </form>

        <c:if test="${!empty listaPrestadores}">
                <table border="1">
                        <tr>
                                <th>CNPJ/CPF</th>
                                <th>Nome</th>
                                <th>Tipo</th>
                                <th>Alterar</th>
                                <th>Excluir</th>
                        </tr>
                        <c:forEach var="prestador" items="${listaPrestadores}">
                                <tr>
                                        <c:if test="${prestador.tipo=='fisico'}">
                                        <td>${prestador.cpf}</td>
                                        <td>${prestador.nome}</td>
                                        <td>${prestador.tipo}</td>
                                        <td><a
href="javascript:editarPrestadorFisico('${prestador.cpf}')"><img
                                                src="img/alterar.png" border=0></a></td>
                <td><a href="javascript:removerPrestador('${prestador.cpf}')"><img
                                                src="img/stop.gif" border=0></a></td>
                                        </c:if>
                                        <c:if test="${prestador.tipo=='juridico'}">
                                        <td>${prestador.cnpj}</td>
                                        <td>${prestador.nome}</td>
                                        <td>${prestador.tipo}</td>
                                        <td><a
href="javascript:editarPrestadorJuridico('${prestador.cnpj}')"><img
                                                src="img/alterar.png" border=0></a></td>
                <td><a href="javascript:removerPrestador('${prestador.cnpj}')"><img
                                                src="img/stop.gif" border=0></a></td>
                                        </c:if>
                                </tr>
                        </c:forEach>
                </table>
        </c:if>
        <form id="form2" name="form2" method="post" action="SvtBeneficiarioCadastro.ts">
                <input type="Submit" name="CadastrarNovoPrestador" value="Novo
Cadastro" onclick="setVisible('popup');return false" target="_self">
        </form>
<div id="popup">
```

```
 <span id="close"><a href="javascript:setVisible('popup')" style="text-decoration: none">
            <strong><input type="submit" name="cancelar" value="Cancelar"
onclick="setVisible('popup')">
                                        </strong>
                                    </a>
 </span>
  <p>
    <form action="SvtPrestadorCadastro" id="formulariocpfcnpj" method="post">
                <div class="linha1">
                <label>CPF ou CNPJ:   <input type="text" id="cpfcnpj" name="cpfcnpj"
value=""/>
                </label>
                </div>
                <div class="linhabotoes">
        <input type="submit" name="Continuar" value="Continuar" onclick="return
validarTamanho();">
                </div>
          </form>
  </p>
  <br><br>
</div>
</body>
</html>
```

Nome do arquivo: pesquisaUsu.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pesquisa :: Usuários</title>
<script language= "JavaScript" src="js/funcoesUsuario.js"></script>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body>
        <c:if test="${!empty alerta}">
                <script language="JavaScript">
                        alert('${alerta}');
                </script>
        </c:if>
        <form id="formEditar" name="formEditar" method="post"
action="SvtUsuarioCadastro.ts">
                <input type=hidden name="operacao" value="editar">
                <input type=hidden name="id" value="">
        </form>
        <form id="formExcluir" name="formExcluir" method="post"
action="SvtUsuarioExcluir.ts">
                <input type=hidden name="id" value="">
        </form>
        <form id="filtro" name="filtro" method="post" action="SvtUsuarioPesquisa.ts"
class="niceform">
        <h3> Pesquisa de Usuários </h3>
        <fieldset><legend>Filtro de Pesquisa</legend>
        <div class="linha"><label>Nome: <input name="nome"
```

```
                type="text" id="nome" size="40" maxlength="40" value="${nome}"> </label>
        </div>
        <input type="Submit" name="pesquisar" value="Pesquisar" /></fieldset>
        </form>
        <c:if test="${!empty listaUsuarios}">
                <table border="1">
                        <tr>
                                <th>Código</th>
                                <th>Nome Usuário</th>
                                <th>Login</th>
                                <th>Ativo</th>
                                <th>Alterar</th>
                                <th>Excluir</th>
                        </tr>
                        <c:forEach var="usuario" items="${listaUsuarios}">
                                <tr>
                                        <td>${usuario.codigo}</td>
                                        <td>${usuario.usuario}</td>
                                        <td>${usuario.login}</td>
                                        <td>${usuario.ativo}</td>
                                        <td><a
href="javascript:editarUsuario(${usuario.codigo})"><img
                                                src="img/alterar.png" border=0></a></td>
                <td><a href="javascript:removerUsuario(${usuario.codigo})"><img
                                                src="img/stop.gif" border=0></a></td>
                                </tr>

                        </c:forEach>
                </table>
        </c:if>
        <form id="form2" name="form2" method="post" action="SvtUsuarioCadastro.ts">
                <input type="Submit" name="cadastrarnovo" value="Novo Cadastro">
        </form>
</body>
</html>
```

Nome do arquivo: principal.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>TissUFPR</title>
<link rel="stylesheet" type="text/css" media="all" href="niceforms-default.css" />
</head>
<body class="niceform" >
</body>
</html>
```

Nome do arquivo: screen.css

```
/* general */
body
        { margin: 15px; background-color: white; color: #FFFFCC; font-family: Tahoma,
Verdana; font-size: 11px; min-width: 760px;}
```

```
table, td, th
        { color: #000000; font-family: Tahoma, Verdana; font-size: 11px; }
img
        { border: 0 }
a, a:link, a:visited, a:active
        { color: #0000aa; text-decoration: none; }
a:hover
        { color: #ff0000; }
h1
        { background-color: #4682B4; color: #FFFFFF; margin: 0; padding: 3px 3px 3px 10px;
font-size: 13px; }
h2
        { background-color: #DBEAF5; color: #000000; margin: 0 0 5px 0; padding: 3px; font-
size: 12px; }
h1 a, h1 a:link, h1 a:visited,
h2 a, h2 a:link, h2 a:visited,
h3 a, h3 a:link, h3 a:visited
        { color: black; }
hr
        { color: #DBEAF5; height : 1px; }
pre
        { background: #dddddd; border: 1px solid black; color: black; padding-top: 1em; white-
space: pre; padding: 10px;}
p
        { margin-left: 10px; }
ul, ol
        { margin-left: 30px; }
form
        { margin: 0; }

/* header */
img#prodLogo
        { width: 350px; height: 80px; float: left; }
img#corpLogo
        { width: 178px; height: 30px; float: right; }
div#slogan
        { margin: 0; white-space: nowrap; float: right; clear: right; font-size: 10px;}
div#topNav
        { margin: 25px 0 0 0; white-space: nowrap; float: right; clear: right; }
div#topNav a
        { font-weight: bold;}

/* content */
div#content
        { border-right: 1px solid #4682B4; background: url('img/leftBG.gif') repeat-y left; }
div#leftPane
        { width: 200px; padding: 0 0 20px 0; float: left; }
div#leftPane h1
        { margin: 20px 0 0 0; padding-left: 10px; }
div#leftPane h2
        { margin: 10px 0 5px 1px; border-bottom: 1px solid #d0d0d0; padding-left: 10px;}
div#leftPane .first
        { margin-top: 0; }
div#leftPane a
        { display: block; margin-left: 10px; }
div#leftPane a.sect
        { font-weight: bold; }
div#rightPane
        { margin-left: 220px; padding: 0 20px 20px 10px; }
div#rightPane h2
```

```
        { margin: 20px -20px 5px -9px; border-bottom: 1px solid #d0d0d0; border-left: 1px solid
#d0d0d0; padding-left: 10px; }
div#rightPane h1
        { margin: 20px -20px 5px -30px; padding-left: 30px; }
div#rightPane .first
        { margin-top: 0; }
table.compat
        { border: 1px solid silver; border-collapse: collapse; margin: 5px auto 5px auto; }
table.compat th
        { border: 1px solid silver; border-collapse: collapse; background-color: #e0e0e0;
padding: 4px 8px 4px 8px; min-width: 60px; }
table.compat td
        { border: 1px solid silver; border-collapse: collapse; padding: 4px; }
.clearing
        { height: 5px; margin: 0; padding: 0; clear: both; font-size: 1px; }
.com
        { color: gray; }

/* footer */
div#footer
        { background-color: #4682B4; color: #FFFFFF; padding: 3px 2px 3px 10px; height:
15px; clear:both; }
div#cpRight
        { float : left; white-space: nowrap; }
div#botmLinks
        { float: right; white-space: nowrap; }
div#footer div#botmLinks
        { color: #000000; white-space: nowrap; }
div#botmLinks a
        { color: #FFFFFF; white-space: nowrap; }
div#botmLinks a:hover
        { text-decoration: underline; }
```

Nome do arquivo: validacaocpf.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="JavaScript" src="validacao.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form>
<input type="text" name="sami" />
<input type="submit" onClick="return valida_cnpj(this.form.sami.value)">
</form>
</body>
</html>
```

Nome do arquivo: funcoesUsuario.js

```
function verificaSenhas() {
        if (document.getElementById('senha').value == document
                        .getElementById('confsenha').value) {
                return true;
```

```
	} else {
		alert('Favor verificar a senha.');
		return false;
	}
}

function editarUsuario(idUsuario) {
	document.getElementById('formEditar').id.value = idUsuario;
	document.getElementById('formEditar').submit();
}

function removerUsuario(idUsuario) {
	if (confirm("Remover Usuario?")) {
		document.getElementById('formExcluir').id.value = idUsuario;

		document.getElementById('formExcluir').submit();
	}
}
```

Nome do arquivo: aba.js

```
function JTabControl(divMaster,tabWidth,tabHeight)
{

	this.tabs = 0;
	this.addTab = addTab;
	this.id  = divMaster;

	Construtor(tabWidth,tabHeight);

	this.Container = getObj(divMaster+".ContainerDiv");
	this.Header = getObj(divMaster+".HeaderDiv");
	this.Body = getObj(divMaster+".BodyDiv");

	function Construtor(tabWidth,tabHeight)
	{
		var idContainer = divMaster+".ContainerDiv";
		var idHeader = divMaster+".HeaderDiv";
		var idBody = divMaster+".BodyDiv";

		this.Container = getObj(divMaster);
		this.Container.innerHTML =  "<div id='"+idContainer+"'
class='jAbaContainer'></div>" + this.Container.innerHTML;
		this.Container = document.getElementById(idContainer);
		this.Container.innerHTML = "<div id='"+idHeader+"'
Class='jAbaHeader'></div>" + "<div id='"+idBody+"' Class='jAbaBody'></div>"
		this.Header = getObj(idHeader);
		this.Body = getObj(idBody);

		this.Container.style.width = tabWidth;
		this.Container.style.height = tabHeight;

	}

	function addTab(divTab,TitleTab)
	{
		var htmlHeader = "";
		var htmlBody = "";
		var objDivOld = getObj(divTab);
```

```
                var bodyClass = "jTabBodyHidden";
                var headerClass = "jTabHeaderHidden";

                if(!TitleTab)
                        TitleTab = divTab;
                if(this.tabs == 0)
                        bodyClass = "jTabBodyInline";
                if(this.tabs == 0)
                        headerClass = "jTabHeaderInline";

                htmlHeader = "<Span onClick='changeJTabControl(this.id)'
id='"+this.id+".Header.Tab."+this.tabs+"'
Class='"+headerClass+"'>"+TitleTab+"</Span> ";
                htmlBody = " <Span id='"+this.id+".Body.Tab."+this.tabs+"'
Class='"+bodyClass+"'>"+objDivOld.innerHTML+"</Tab>";
                objDivOld.innerHTML = "";

                Header.innerHTML = Header.innerHTML + htmlHeader;
                Body.innerHTML = Body.innerHTML + htmlBody;

                this.tabs++;
        }
}

function changeJTabControl(Tab)
{
        var id = Tab.split(".");
        var i = 0;
        var j = 0;

        while(getObj(id[0]+".Header.Tab."+i))
        {
                getObj(id[0]+".Header.Tab."+i).className = "jTabHeaderHidden"
                i++;
        }
        getObj(Tab).className = "jTabHeaderInline";

        i=0;
        while(getObj(id[0]+".Body.Tab."+j))
        {
                getObj(id[0]+".Body.Tab."+j).className = "jTabBodyHidden"
                j++;
        }
        getObj(id[0]+".Body.Tab."+id[3]).className = "jTabBodyInline";

}
function getObj(idObj)
{
        return document.getElementById(idObj);
}
```

Nome do arquivo: funcoesBeneficiario.js

```
function editarBeneficiario(idBeneficiario) {
        document.getElementById('formEditar').id.value = idBeneficiario;
        document.getElementById('formEditar').submit();
}

function removerBeneficiario(idBeneficiario) {
```

```
        if (confirm("Remover Beneficario?")) {
                document.getElementById('formExcluir').id.value = idBeneficiario;

                document.getElementById('formExcluir').submit();
        }
}
```

Nome do arquivo: funcoesEndereco.js

```
function editarEndereco(cpfcnpj, codigo) {
        document.getElementById('formEditar').cpfcnpj.value = cpfcnpj;
        document.getElementById('formEditar').codigo.value = codigo;
        document.getElementById('formEditar').submit();
}

function removerEndereco(cpfcnpj, codigo) {
        if (confirm("Remover Endereco?")) {
                document.getElementById('formExcluir').cpfcnpj.value = cpfcnpj;
                document.getElementById('formExcluir').codigo.value = codigo;
                document.getElementById('formExcluir').submit();
        }
}
```

Nome do arquivo: funcoesGuia.js

```
        function adicionar(elemento, max) {
                div = document.getElementById(elemento);
                parente = div.parentNode;
                if (parente.childNodes.length-1 < max) {
                        last = parente.lastChild;
                        clone = last.cloneNode(true);
                        inputs = clone.getElementsByTagName("input");
                        for (i=0; i < inputs.length; i++) {
                                inputs[i].value = '';
                        }
                        parente.insertBefore(clone, null);
                }
        }

        function adicionar(elemento) {
                div = document.getElementById(elemento);
                parente = div.parentNode;
                last = parente.lastChild;
                clone = last.cloneNode(true);
                inputs = clone.getElementsByTagName("input");
                for (i=0; i < inputs.length; i++) {
                        inputs[i].value = '';
                }
                parente.insertBefore(clone, null);
        }

        function visi(nr){
          if (document.layers){
                vista = (document.layers[nr].visibility == 'hide') ? 'show' : 'hide'
                document.layers[nr].visibility = vista;
          }
          else if (document.all){
                vista = (document.all[nr].style.visibility == 'hidden') ? 'visible'      : 'hidden';
```

```
                document.all[nr].style.visibility = vista;
            }
            else if (document.getElementById){
                vista = (document.getElementById(nr).style.visibility == 'hidden') ? 'visible' :
'hidden';
                document.getElementById(nr).style.visibility = vista;
            }
        }

        function blocking(nr){
            if (document.layers){
                current = (document.layers[nr].display == 'none') ? 'block' : 'none';
                document.layers[nr].display = current;
            }
            else if (document.all){
                current = (document.all[nr].style.display == 'none') ? 'block' : 'none';
                document.all[nr].style.display = current;
            }
            else if (document.getElementById){
                        vista = (document.getElementById(nr).style.display == 'none') ? 'block'
: 'none';
                        document.getElementById(nr).style.display = vista;
            }

            if (document.getElementById("hdimg"+ nr).value == 0){
                document.getElementById("img"+ nr).src = "img/actionhidefilters.gif";
              document.getElementById("hdimg"+ nr).value ="1";
            } else {
                document.getElementById("img"+ nr).src = "img/actionshowfilters.gif";
              document.getElementById("hdimg"+ nr).value ="0";
            }
        }
        function inserirEmCamposNumericos(texto, left, top) {
                for (i=0; i < texto.length; i++) {
                        carater = texto.substring(i, i+1);
                        var newdiv = document.getElementById('template').cloneNode(true);
                        newdiv.innerHTML = carater;
                        newdiv.style.top = top + 'px';
                        newdiv.style.left = (left + (i * 23.2)) + 'px';
                        document.body.appendChild(newdiv);
                }
        }

        function inserirEmCamposData(texto, left, top) {
                incrementos = new Array(6);
                incrementos[0] = 23.2;
                incrementos[1] = 36;
                incrementos[2] = 23.2;
                incrementos[3] = 36;
                incrementos[4] = 23.2;
                incrementos[5] = 0;

                posicoes = new Array(6);
                posicoes[0] = 0;
                posicoes[1] = 1;
                posicoes[2] = 3;
                posicoes[3] = 4;
                posicoes[4] = 8;
                posicoes[5] = 9;
```

```
                for (i=0; i < 6; i++) {
                        carater = texto.substring(posicoes[i], posicoes[i]+1);
                        var newdiv = document.getElementById('template').cloneNode(true);
                        newdiv.innerHTML = carater;
                        newdiv.style.top = top + 'px';
                        newdiv.style.left = left + 'px';
                        left += incrementos[i];
                        document.body.appendChild(newdiv);
                }
        }
```

Nome do arquivo: funcoesOperadora.js

```
function editarOperadora(idOperadora) {
        document.getElementById('formEditar').id.value = idOperadora;
        document.getElementById('formEditar').submit();
}

function removerOperadora(idOperadora) {
        if (confirm("Remover Operadora?")) {
                document.getElementById('formExcluir').id.value = idOperadora;

                document.getElementById('formExcluir').submit();
        }
}
function selecionar_tudo(nomeGuia){
          for (i=0;i<document.cadastro.elements.length;i++) {
                    if(document.cadastro.elements[i].id == nomeGuia)
               document.cadastro.elements[i].checked=1
          }
}
function deselecionar_tudo(nomeGuia){
          for (i=0;i<document.cadastro.elements.length;i++) {
                    if(document.cadastro.elements[i].id == nomeGuia)
               document.cadastro.elements[i].checked=0
          }
}
```

Nome do arquivo: funcoesPrestador.js

```
function editarPrestadorFisico(cpf) {
        document.getElementById('formEditarFisico').cpf.value = cpf;
        document.getElementById('formEditarFisico').submit();
}

function editarPrestadorJuridico(cnpj) {
        document.getElementById('formEditarJuridico').cnpj.value = cnpj;
        document.getElementById('formEditarJuridico').submit();
}

function removerPrestador(idPrestador) {
        if (confirm("Remover Prestador?")) {

                document.getElementById('formExcluir').id.value = idPrestador;

                document.getElementById('formExcluir').submit();
        }
}
```

```
function valida_cpf(cpf) {
        var numeros, digitos, soma, i, resultado, digitos_iguais;
        digitos_iguais = 1;
        if (cpf.length < 11) {
                alert("Informe 11 digitos.");
                return false;
        }
        for (i = 0; i < cpf.length - 1; i++) {
                if (cpf.charAt(i) != cpf.charAt(i + 1)) {
                        digitos_iguais = 0;
                        break;
                }
        }
        if (!digitos_iguais) {
                numeros = cpf.substring(0, 9);
                digitos = cpf.substring(9);
                soma = 0;
                for (i = 10; i > 1; i--) {
                        soma += numeros.charAt(10 - i) * i;
                }

                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(0)) {
                        alert("CPF inválido");
                        return false;
                }
                numeros = cpf.substring(0, 10);
                soma = 0;
                for (i = 11; i > 1; i--) {
                        soma += numeros.charAt(11 - i) * i;
                }

                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(1)) {
                        alert("CPF inválido");
                        return false;
                }
                return true;
        } else {
                alert("CPF inválido");
                return false;
        }

}

function valida_cnpj(cnpj) {
        var numeros, digitos, soma, i, resultado, pos, tamanho, digitos_iguais;
        digitos_iguais = 1;
        if (cnpj.length < 14 && cnpj.length < 15) {
                alert("CNPJ inválido.");
                return false;
        }

        for (i = 0; i < cnpj.length - 1; i++) {
                if (cnpj.charAt(i) != cnpj.charAt(i + 1)) {
                        digitos_iguais = 0;
                        break;
                }
        }
```

```
        if (!digitos_iguais) {
                tamanho = cnpj.length - 2
                numeros = cnpj.substring(0, tamanho);
                digitos = cnpj.substring(tamanho);
                soma = 0;
                pos = tamanho - 7;
                for (i = tamanho; i >= 1; i--) {
                        soma += numeros.charAt(tamanho - i) * pos--;
                        if (pos < 2) {
                                pos = 9;
                        }
                }
                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(0)) {
                        alert("CNPJ inválido.");
                        return false;
                }

                tamanho = tamanho + 1;
                numeros = cnpj.substring(0, tamanho);
                soma = 0;
                pos = tamanho - 7;
                for (i = tamanho; i >= 1; i--) {
                        soma += numeros.charAt(tamanho - i) * pos--;
                        if (pos < 2) {
                                pos = 9;
                        }
                }
                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(1)) {
                        alert("CNPJ inválido.");
                        return false;
                }
                return true;
        } else {
                return false;
        }
}


function validarTamanho() {
        cpfcnpj = document.getElementById('cpfcnpj').value;

        if (cpfcnpj.length == 11) {
                if (valida_cpf(cpfcnpj)) {
        document.getElementById('formulariocpfcnpj').action="SvtPrestadorFisicoCadastro.ts";
                        return true;
                }
        } else if (cpfcnpj.length == 14) {
                if (valida_cnpj(cpfcnpj)) {
        document.getElementById('formulariocpfcnpj').action="SvtPrestadorJuridicoCadastro.ts
";
                        return true;
                }
        } else {
                alert("CPF / CNPJ invalido.");
                return false;
        }
}
```

```
/* ----------------------------------------------
Floating layer - v.1
(c) 2006 www.haan.net
contact: jeroen@haan.net
You may use this script but please leave the credits on top intact.
Please inform us of any improvements made.
When usefull we will add your credits.
---------------------------------------------- */

x = 20;
y = 70;
function setVisible(obj)
{
        obj = document.getElementById(obj);
        obj.style.visibility = (obj.style.visibility == 'visible') ? 'hidden' : 'visible';
}
function placeIt(obj)
{
        obj = document.getElementById(obj);
        if (document.documentElement)
        {
                theLeft = document.documentElement.scrollLeft;
                theTop = document.documentElement.scrollTop;
        }
        else if (document.body)
        {
                theLeft = document.body.scrollLeft;
                theTop = document.body.scrollTop;
        }
        theLeft += x;
        theTop += y;
        obj.style.left = theLeft + 'px' ;
        obj.style.top = theTop + 'px' ;
        setTimeout("placeIt('popup')",500);
}
```

Nome do arquivo: validacao.js

```
function valida_cpf(cpf) {
        var numeros, digitos, soma, i, resultado, digitos_iguais;
        digitos_iguais = 1;
        if (cpf.length < 11) {
                alert("Informe 11 digitos.");
                return false;
        }
        for (i = 0; i < cpf.length - 1; i++) {
                if (cpf.charAt(i) != cpf.charAt(i + 1)) {
                        digitos_iguais = 0;
                        break;
                }
        }
        if (!digitos_iguais) {
                numeros = cpf.substring(0, 9);
                digitos = cpf.substring(9);
                soma = 0;
                for (i = 10; i > 1; i--) {
                        soma += numeros.charAt(10 - i) * i;
                }
```

```
                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(0)) {
                        alert("CPF inválido");
                        return false;
                }
                numeros = cpf.substring(0, 10);
                soma = 0;
                for (i = 11; i > 1; i--) {
                        soma += numeros.charAt(11 - i) * i;
                }

                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(1)) {
                        alert("CPF inválido");
                        return false;
                }
                return true;
        } else {
                alert("CPF inválido");
                return false;
        }

}

function valida_cnpj(cnpj) {
        var numeros, digitos, soma, i, resultado, pos, tamanho, digitos_iguais;
        digitos_iguais = 1;
        if (cnpj.length < 14 && cnpj.length < 15) {
                alert("CNPJ inválido.");
                return false;
        }

        for (i = 0; i < cnpj.length - 1; i++) {
                if (cnpj.charAt(i) != cnpj.charAt(i + 1)) {
                        digitos_iguais = 0;
                        break;
                }
        }

        if (!digitos_iguais) {
                tamanho = cnpj.length - 2
                numeros = cnpj.substring(0, tamanho);
                digitos = cnpj.substring(tamanho);
                soma = 0;
                pos = tamanho - 7;
                for (i = tamanho; i >= 1; i--) {
                        soma += numeros.charAt(tamanho - i) * pos--;
                        if (pos < 2) {
                                pos = 9;
                        }
                }
                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(0)) {
                        alert("CNPJ inválido.");
                        return false;
                }

                tamanho = tamanho + 1;
                numeros = cnpj.substring(0, tamanho);
```

```
                soma = 0;
                pos = tamanho - 7;
                for (i = tamanho; i >= 1; i--) {
                        soma += numeros.charAt(tamanho - i) * pos--;
                        if (pos < 2) {
                                pos = 9;
                        }
                }
                resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
                if (resultado != digitos.charAt(1)) {
                        alert("CNPJ inválido.");
                        return false;
                }
                return true;
        } else {
                return false;
        }
}
```

Nome do arquivo: web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:jsp="http://java.sun.com/xml/ns/javaee/jsp"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" id="WebApp_ID" version="2.5">
 <display-name>TissUFPR</display-name>
 <welcome-file-list>
  <welcome-file>login.jsp</welcome-file>
 </welcome-file-list>
 <servlet>
  <description></description>
  <display-name>SvtLogin</display-name>
  <servlet-name>SvtLogin</servlet-name>
  <servlet-class>com.servlets.SvtLogin</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtLogin</servlet-name>
  <url-pattern>/login.jsp</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtUsuarioPesquisa</display-name>
  <servlet-name>SvtUsuarioPesquisa</servlet-name>
  <servlet-class>com.servlets.usuario.Pesquisa</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtUsuarioPesquisa</servlet-name>
  <url-pattern>/SvtUsuarioPesquisa.ts</url-pattern>
 </servlet-mapping>
 <jsp-config>
  <taglib>
   <taglib-uri>http://java.sun.com/jstl/core</taglib-uri>
   <taglib-location>/WEB-INF/c.tld</taglib-location>
  </taglib>
 </jsp-config>
 <servlet>
  <description></description>
```

```
  <display-name>SvtOperadoraCadastro</display-name>
  <servlet-name>SvtOperadoraCadastro</servlet-name>
  <servlet-class>com.servlets.operadora.Cadastro</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtOperadoraCadastro</servlet-name>
  <url-pattern>/SvtOperadoraCadastro.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtPrestadorPesquisa</display-name>
  <servlet-name>SvtPrestadorPesquisa</servlet-name>
  <servlet-class>com.servlets.prestador.Pesquisa</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtPrestadorPesquisa</servlet-name>
  <url-pattern>/SvtPrestadorPesquisa.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtBeneficiarioCadastro</display-name>
  <servlet-name>SvtBeneficiarioCadastro</servlet-name>
  <servlet-class>com.servlets.beneficiario.Cadastro</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtBeneficiarioCadastro</servlet-name>
  <url-pattern>/SvtBeneficiarioCadastro.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtUsuarioCadastro</display-name>
  <servlet-name>SvtUsuarioCadastro</servlet-name>
  <servlet-class>com.servlets.usuario.Cadastro</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtUsuarioCadastro</servlet-name>
  <url-pattern>/SvtUsuarioCadastro.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtUsuarioExcluir</display-name>
  <servlet-name>SvtUsuarioExcluir</servlet-name>
  <servlet-class>com.servlets.usuario.Excluir</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtUsuarioExcluir</servlet-name>
  <url-pattern>/SvtUsuarioExcluir.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtBeneficiarioPesquisa</display-name>
  <servlet-name>SvtBeneficiarioPesquisa</servlet-name>
  <servlet-class>com.servlets.beneficiario.Pesquisa</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtBeneficiarioPesquisa</servlet-name>
  <url-pattern>/SvtBeneficiarioPesquisa.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
```

```
  <display-name>SvtBeneficiarioExcluir</display-name>
  <servlet-name>SvtBeneficiarioExcluir</servlet-name>
  <servlet-class>com.servlets.beneficiario.Excluir</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtBeneficiarioExcluir</servlet-name>
  <url-pattern>/SvtBeneficiarioExcluir.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtOperadoraPesquisa</display-name>
  <servlet-name>SvtOperadoraPesquisa</servlet-name>
  <servlet-class>com.servlets.operadora.Pesquisa</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtOperadoraPesquisa</servlet-name>
  <url-pattern>/SvtOperadoraPesquisa.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtPrestadorFisicoCadastro</display-name>
  <servlet-name>SvtPrestadorFisicoCadastro</servlet-name>
  <servlet-class>com.servlets.prestador.fisico.Cadastro</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtPrestadorFisicoCadastro</servlet-name>
  <url-pattern>/SvtPrestadorFisicoCadastro.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtPrestadorJuridicoCadastro</display-name>
  <servlet-name>SvtPrestadorJuridicoCadastro</servlet-name>
  <servlet-class>com.servlets.prestador.juridico.Cadastro</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtPrestadorJuridicoCadastro</servlet-name>
  <url-pattern>/SvtPrestadorJuridicoCadastro.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtOperadoraExcluir</display-name>
  <servlet-name>SvtOperadoraExcluir</servlet-name>
  <servlet-class>com.servlets.operadora.Excluir</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtOperadoraExcluir</servlet-name>
  <url-pattern>/SvtOperadoraExcluir.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtPrestadorExcluir</display-name>
  <servlet-name>SvtPrestadorExcluir</servlet-name>
  <servlet-class>com.servlets.prestador.Excluir</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtPrestadorExcluir</servlet-name>
  <url-pattern>/SvtPrestadorExcluir.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
```

```
  <display-name>SvtEscolheOperadora</display-name>
  <servlet-name>SvtEscolheOperadora</servlet-name>
  <servlet-class>com.servlets.guia.EscolheOperadora</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtEscolheOperadora</servlet-name>
  <url-pattern>/SvtEscolheOperadora.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtEscolheBeneficiario</display-name>
  <servlet-name>SvtEscolheBeneficiario</servlet-name>
  <servlet-class>com.servlets.guia.EscolheBeneficiario</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtEscolheBeneficiario</servlet-name>
  <url-pattern>/SvtEscolheBeneficiario.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtEscolhePrestador</display-name>
  <servlet-name>SvtEscolhePrestador</servlet-name>
  <servlet-class>com.servlets.guia.EscolhePrestador</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtEscolhePrestador</servlet-name>
  <url-pattern>/SvtEscolhePrestador.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtPrePreenchimentoGuia</display-name>
  <servlet-name>SvtPrePreenchimentoGuia</servlet-name>
  <servlet-class>com.servlets.guia.PrePreenchimentoGuia</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtPrePreenchimentoGuia</servlet-name>
  <url-pattern>/SvtPrePreenchimentoGuia.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>SvtImprimirGuiaConsulta</display-name>
  <servlet-name>SvtImprimirGuiaConsulta</servlet-name>
  <servlet-class>com.servlets.guia.ImprimirGuiaConsulta</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>SvtImprimirGuiaConsulta</servlet-name>
  <url-pattern>/SvtImprimirGuiaConsulta.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>ServletEnderecoPesq</display-name>
  <servlet-name>ServletEnderecoPesq</servlet-name>
  <servlet-class>com.servlets.prestador.ServletEnderecoPesq</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ServletEnderecoPesq</servlet-name>
  <url-pattern>/ServletEnderecoPesq.ts</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
```

```
  <display-name>SvtEndereco</display-name>
  <servlet-name>SvtEndereco</servlet-name>
  <servlet-class>com.servlets.prestador.SvtEndereco</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtEndereco</servlet-name>
  <url-pattern>/SvtEndereco.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>svtGuia</display-name>
  <servlet-name>svtGuia</servlet-name>
  <servlet-class>com.servlets.guia.svtGuia</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>svtGuia</servlet-name>
  <url-pattern>/svtGuia.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtConfig</display-name>
  <servlet-name>SvtConfig</servlet-name>
  <servlet-class>com.servlets.SvtConfig</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtConfig</servlet-name>
  <url-pattern>/SvtConfig.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>SvtImprimeGuiaXML</display-name>
  <servlet-name>SvtImprimeGuiaXML</servlet-name>
  <servlet-class>com.servlets.guia.ImprimeGuiaXML</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>SvtImprimeGuiaXML</servlet-name>
  <url-pattern>/SvtImprimeGuiaXML.ts</url-pattern>
 </servlet-mapping>
 <servlet>
  <description></description>
  <display-name>svtConsultarGuia</display-name>
  <servlet-name>svtConsultarGuia</servlet-name>
  <servlet-class>com.servlets.guia.svtConsultarGuia</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>svtConsultarGuia</servlet-name>
  <url-pattern>/svtConsultarGuia.ts</url-pattern>
 </servlet-mapping>
</web-app>
```

Classes

Pacote com.classes

Nome do arquivo: ClBeneficiario.java

```
package com.classes;

import com.DAO.DAOBeneficiario;
```

```
public class CIBeneficiario {
	private Integer codigo;
	private String nome;
	private String numcarteira;
	private String valcarteira;
	private String tpplano;
	private String numcartaonac;

	public void setNome(String nome) {
		this.nome = nome;
	}
	public String getNome() {
		return nome;
	}
	public void setNumcarteira(String numcarteira) {
		this.numcarteira = numcarteira;
	}
	public String getNumcarteira() {
		return numcarteira;
	}
	public void setValcarteira(String valcarteira) {
		this.valcarteira = valcarteira;
	}
	public String getValcarteira() {
		return valcarteira;
	}
	public void setTpplano(String tpplano) {
		this.tpplano = tpplano;
	}
	public String getTpplano() {
		return tpplano;
	}
	public void setNumcartaonac(String numcartaonac) {
		this.numcartaonac = numcartaonac;
	}
	public String getNumcartaonac() {
		return numcartaonac;
	}
	public void setCodigo(Integer codigo) {
		this.codigo = codigo;
	}
	public Integer getCodigo() {
		return codigo;
	}

	public void cadastrarBeneficiario(CIBeneficiario beneficiario){
		DAOBeneficiario daobeneficiario = new DAOBeneficiario();
		daobeneficiario.cadastrarBeneficiario(beneficiario);
	}
	public void removerBeneficiario(int codigo){
		DAOBeneficiario daobeneficiario = new DAOBeneficiario();
		daobeneficiario.removerBeneficiario(codigo);
	}
	public CIBeneficiario obterDetalhesBeneficiario(int codigo){
		DAOBeneficiario daobeneficiario = new DAOBeneficiario();
		CIBeneficiario beneficiario     = new CIBeneficiario();
		beneficiario                    = daobeneficiario.obterDetalhesBeneficiario(codigo);
		return beneficiario;
	}
```

```
	public void editarBeneficiario(CIBeneficiario beneficiario){
		DAOBeneficiario daobeneficiario = new DAOBeneficiario();
		daobeneficiario.editarBeneficiario(beneficiario);
	}
}
```

Nome do arquivo: CICampoGuia.java

```
package com.classes;

public class CICampoGuia {
	private boolean checked;
	private String descricao;
	private String nomeCompleto;

	public String getDescricao() {
		return descricao;
	}
	public void setDescricao(String descricao) {
		this.descricao = descricao;
	}
	public String getNomeCompleto() {
		return nomeCompleto;
	}
	public void setNomeCompleto(String nomeCompleto) {
		this.nomeCompleto = nomeCompleto;
	}

	public boolean isChecked() {
		return checked;
	}
	public void setChecked(boolean checked) {
		this.checked = checked;
	}
}
```

Nome do arquivo: CICBOS.java

```
package com.classes;

public class CICBOS {
	private String cBOS;
	private String descricao;

	public void setCBOS(String CBOS){
		this.cBOS = CBOS;
	}
	public String getCBOS(){
		return this.cBOS;
	}
	public void setPlano(String Descricao){
		this.descricao = Descricao;
	}
	public String getDescricao(){
		return this.descricao;
	}
}
```

Nome do arquivo: CIConfig.java

```
package com.classes;

import com.DAO.DAOConfig;

public class CIConfig {

	private String complexType = "";
	private String dirGuias    = "";

	public void setComplexType(String complexType){
		this.complexType = complexType;
	}
	public String getComplexType(){
		return this.complexType;
	}
	public void setDirGuias(String dirGuias){
		this.dirGuias = dirGuias;
	}
	public String getDirGuias(){
		return this.dirGuias;
	}

	public CIConfig buscaConfig(){
		DAOConfig dAOConfig = new DAOConfig();
		CIConfig clConfig   = dAOConfig.buscaConfig();
		return clConfig;
	}
	public boolean editarConfig(CIConfig config) {
		boolean alterou     = true;
		DAOConfig dAOConfig = new DAOConfig();
		alterou             = dAOConfig.editarConfig(config);
		return alterou;
	}
}
```

Nome do arquivo: CIConselhoProfissional.java

```
package com.classes;

public class CIConselhoProfissional {

	private String numeroConselhoProf;
	private String descricaoConselhoProf;

	public void setNumeroConselhoProf(String numeroConselhoProf){
		this.numeroConselhoProf = numeroConselhoProf;
	}
	public String getNumeroConselhoProf(){
		return this.numeroConselhoProf;
	}
	public void setDescricaoConselhoProf(String descricaoConselhoProf){
		this.descricaoConselhoProf = descricaoConselhoProf;
	}
	public String getDescricaoConselhoProf(){
		return this.descricaoConselhoProf;
	}
```

```
}
```

Nome do arquivo: ClEndereco.java

```
package com.classes;

import com.DAO.DAOEndereco;

public class ClEndereco {
	private String cFPCNPJ;
	private int    codigo;
	private String tipoLogradouro;
	private String logradouro;
	private String numero;
	private String complemento;
	private String cep;
	private String municipio;
	private String uf;
	private String ibge;
	private String principal;

	public ClEndereco(){
		this.codigo = 0;
	}
	public void setLogradouro(String logradouro) {
		this.logradouro = logradouro;
	}
	public String getLogradouro() {
		return logradouro;
	}
	public void setNumero(String numero) {
		this.numero = numero;
	}
	public String getNumero() {
		return numero;
	}
	public void setComplemento(String complemento) {
		this.complemento = complemento;
	}
	public String getComplemento() {
		return complemento;
	}
	public void setCep(String cep) {
		this.cep = cep;
	}
	public String getCep() {
		return cep;
	}
	public void setMunicipio(String municipio) {
		this.municipio = municipio;
	}
	public String getMunicipio() {
		return municipio;
	}
	public void setUf(String uf) {
		this.uf = uf;
	}
	public String getUf() {
		return uf;
```

```
}
public void setIbge(String ibge) {
        this.ibge = ibge;
}
public String getIbge() {
        return ibge;
}
public void setCFPCNPJ(String cpfcnpj) {
        this.cFPCNPJ = cpfcnpj;
}
public String getCFPCNPJ() {
        return cFPCNPJ;
}
public int getCodigo(){
        return this.codigo;
}
public void setCodigo(int codigo) {
        this.codigo = codigo;
}
public String getTipoLogradouro(){
        return this.tipoLogradouro;
}
public void setTipoLogradouro(String tipoLogradouro) {
        this.tipoLogradouro = tipoLogradouro;
}
public void setPrincipal(String principal) {
        this.principal = principal;
}
public String getPrincipal() {
        return this.principal;
}

public void excluirEndereco(String cpfcnpj, int codigo){
        DAOEndereco dAOEndereco = new DAOEndereco();
        dAOEndereco.excluirEndereco(cpfcnpj, codigo);
}
public void cadastrarEndereco(ClEndereco clEndereco){
        DAOEndereco dAOEndereco = new DAOEndereco();
        dAOEndereco.cadastrarEndereco(clEndereco);
}
public ClEndereco buscaEndereco(String cpfcnpj, int codigo){
        DAOEndereco dAOEndereco = new DAOEndereco();
        ClEndereco clEndereco   = new ClEndereco();
        clEndereco              = dAOEndereco.buscaEndereco(cpfcnpj, codigo);
        return clEndereco;
}
public void atualizarEndereco(ClEndereco clEndereco){
        DAOEndereco dAOEndereco = new DAOEndereco();
        dAOEndereco.atualizarEndereco(clEndereco);
}
public ClEndereco buscaEnderecoPrincipal(String cpfcnpj){
        DAOEndereco dAOEndereco = new DAOEndereco();
        ClEndereco clEndereco   = new ClEndereco();
        clEndereco              = dAOEndereco.buscaEnderecoPrincipal(cpfcnpj);
        return clEndereco;
}
public String buscaTipoLogCod(String dsc){
        DAOEndereco dAOEndereco  = new DAOEndereco();
        String tipoLogradouroCod = dAOEndereco.buscaTipoLogCod(dsc);
        return tipoLogradouroCod;
```

```
        }
}
```

Nome do arquivo: CIGuia.java

```
package com.classes;

import com.DAO.DAOGuia;
import com.util.guia.GeradorFormularioHTMLGuias;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Map;

import java.util.Date;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.validacao.Validacao;

public class CIGuia {

        private String nomeGuia          = "";
        private String complexType       = "";
        private PrintWriter formularioHTML = null;
        private String arquivoXMLString   = "";
        private CIBeneficiario clBeneficiario;
        private CIOperadora clOperadora;
        private Long numeroGuia;
        private String hash;
        private Date dataEmissao;

        public String getNomeGuia(){
                return this.nomeGuia;
        }
        public void setNomeGuia(String nomeGuia){
                this.nomeGuia = nomeGuia;
        }
        public String getComplexType(){
                return this.complexType;
        }
        public void setComplexType(String complexType){
                this.complexType = complexType;
        }
        public PrintWriter getGuiaFormularioHTML(){
                return this.formularioHTML;
        }

        public void setGuiaFormularioHTML(CIGuia clguia, Map<String, String[]> dados,
Map<String, Validacao> validacao){
                GeradorFormularioHTMLGuias gerador = new
GeradorFormularioHTMLGuias();
                this.formularioHTML  = gerador.gerarFormularioHTML(clguia, dados,
validacao);
        }
        public void setGuiaFormularioHTML(CIGuia clguia, HttpServletRequest request){
```

```
                GeradorFormularioHTMLGuias gerador = new
GeradorFormularioHTMLGuias();
                this.formularioHTML  = gerador.gerarFormularioHTML(clguia, request);
        }
        public void setGuiaFormularioHTML(HttpServletResponse response) throws
IOException{
                this.formularioHTML  = response.getWriter();
        }

        public void setGuiaXML(String XMLString){
                this.arquivoXMLString = XMLString;
        }

        public void gravaGuiaXML(String diretorio){
                try {
            BufferedWriter out = new BufferedWriter(new FileWriter(diretorio));
            out.write(this.arquivoXMLString);
            out.close();
          } catch (IOException e) {
                e.printStackTrace();
          }
        }

        public void setClOperadora(ClOperadora clOperadora){
                this.clOperadora = clOperadora;
        }
        public ClOperadora getClOperadora(){
                return this.clOperadora;
        }

        public void setClBeneficiario(ClBeneficiario clBeneficiario){
                this.clBeneficiario = clBeneficiario;
        }
        public ClBeneficiario getClBeneficiario(){
                return this.clBeneficiario;
        }

        public void setNumeroGuia(Long numeroGuia){
                this.numeroGuia = numeroGuia;
        }
        public Long getNumeroGuia(){
                return this.numeroGuia;
        }
        public String getHash(){
                return this.hash;
        }
        public void setHash(String hash){
                this.hash = hash;
        }
        public void setDataEmissao(java.util.Date data){
                this.dataEmissao = data;
        }
        public java.util.Date getDataEmissao(){
                return this.dataEmissao;
        }

        public void cadastrarGuiaEmitida(ClGuia clGuia){
                DAOGuia dAOGuia = new DAOGuia();
                dAOGuia.cadastrarGuiaEmitida(clGuia);
        }
```

```
	public String buscaCtGuia(int codigooperadora, int codbene, String hash){
		DAOGuia dAOGuia = new DAOGuia();
		String ctGuia   = dAOGuia.buscaCtGuia(codigooperadora, codbene, hash);
		return ctGuia;
	}
}
```

Nome do arquivo: ClGuiaConsulta.java

```
package com.classes;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import com.util.guia.Util;

public class ClGuiaConsulta {
	private String STRregistroANS;
	private String STRdataEmissaoGuia;
	private String STRnumeroGuiaPrestador;
	private String STRnumeroGuiaOperadora;

	private String STRnumeroCarteira;
	private String STRnomeBeneficiario;
	private String STRnomePlano;
	private String STRvalidadeCarteira;
	private String STRnumeroCNS;
	private String STRidentificadorBeneficiario;

	private String STRCNPJ;
	private String STRCPF;
	private String STRcodigoPrestadorNaOperadora;
	private String STRnomeContratado;
	private String STRtipoLogradouro;
	private String STRlogradouro;
	private String STRnumero;
	private String STRcomplemento;
	private String STRcodigoIBGEMunicipio;
	private String STRmunicipio;
	private String STRcodigoUF;
	private String STRcep;
	private String STRnumeroCNES;

	private String STRnomeProfissional;
	private String STRsiglaConselho;
	private String STRnumeroConselho;
	private String STRufConselho;
	private String STRcbos;

	// Hipotese Diagnostica
	private String STRdataAtendimento;
	private String STRcodigoTabela;
	private String STRcodigoProcedimento;
	private String STRtipoConsulta;
	private String STRtipoSaida;
	private String STRindicadorAcidente;
```

```
        private String STnomeTabela;
        private String STcodigoDiagnostico;
        private String STdescricaoDiagnostico;
        private String STnomeTabelaSec;
        private String[] STcodigoDiagnosticoSec;
        private String STdescricaoDiagnosticoSec;
        private String Stvalor;
        private String STunidadeTempo;
        private String STtipoDoenca;

        private String STRobservacao;

        public static ClGuiaConsulta getGuiaDoRequest(HttpServletRequest request) {
                ClGuiaConsulta guia = new ClGuiaConsulta();

                guia.STRregistroANS      =
request.getParameter("guiaConsulta.identificacaoGuia.registroANS");
                guia.STRdataEmissaoGuia  =
request.getParameter("guiaConsulta.identificacaoGuia.dataEmissaoGuia");
                guia.STRnumeroGuiaPrestador =
request.getParameter("guiaConsulta.identificacaoGuia.numeroGuiaPrestador");
                guia.STRnumeroGuiaOperadora =
request.getParameter("guiaConsulta.identificacaoGuia.numeroGuiaOperadora");

                guia.STRnumeroCarteira       =
request.getParameter("guiaConsulta.beneficiario.numeroCarteira");
                guia.STRnomeBeneficiario     =
request.getParameter("guiaConsulta.beneficiario.nomeBeneficiario");
                guia.STRnomePlano            =
request.getParameter("guiaConsulta.beneficiario.nomePlano");
                guia.STRvalidadeCarteira     =
request.getParameter("guiaConsulta.beneficiario.validadeCarteira");
                guia.STRnumeroCNS            =
request.getParameter("guiaConsulta.beneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario =
request.getParameter("guiaConsulta.beneficiario.identificadorBeneficiario");

                guia.STRCNPJ                 =
request.getParameter("guiaConsulta.dadosContratado.identificacao.CNPJ");
                guia.STRCPF                  =
request.getParameter("guiaConsulta.dadosContratado.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadora =
request.getParameter("guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperad
ora");
                guia.STRnomeContratado       =
request.getParameter("guiaConsulta.dadosContratado.nomeContratado");
                guia.STRtipoLogradouro       =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.tipoLogradouro");
                guia.STRlogradouro           =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.logradouro");
                guia.STRnumero               =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.numero");
                guia.STRcomplemento          =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.complemento");
                guia.STRcodigoIBGEMunicipio  =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.codigoIBGEMunici
pio");
                guia.STRmunicipio            =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.municipio");
```

```
                guia.STRcodigoUF                =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.codigoUF");
                guia.STRcep                     =
request.getParameter("guiaConsulta.dadosContratado.enderecoContratado.cep");
                guia.STRnumeroCNES              =
request.getParameter("guiaConsulta.dadosContratado.numeroCNES");

                guia.STRnomeProfissional =
request.getParameter("guiaConsulta.profissionalExecutante.nomeProfissional");
                guia.STRsiglaConselho    =
request.getParameter("guiaConsulta.profissionalExecutante.conselhoProfissional.siglaConselh
o");
                guia.STRnumeroConselho   =
request.getParameter("guiaConsulta.profissionalExecutante.conselhoProfissional.numeroCons
elho");
                guia.STRufConselho       =
request.getParameter("guiaConsulta.profissionalExecutante.conselhoProfissional.ufConselho")
;
                guia.STRcbos             =
request.getParameter("guiaConsulta.profissionalExecutante.cbos");

                // Hipotese Diagnostica
                guia.STRdataAtendimento  =
request.getParameter("guiaConsulta.dadosAtendimento.dataAtendimento");
                guia.STRcodigoTabela     =
request.getParameter("guiaConsulta.dadosAtendimento.procedimento.codigoTabela");
                guia.STRcodigoProcedimento =
request.getParameter("guiaConsulta.dadosAtendimento.procedimento.codigoProcedimento");
                guia.STRtipoConsulta     =
request.getParameter("guiaConsulta.dadosAtendimento.tipoConsulta");
                guia.STRtipoSaida        =
request.getParameter("guiaConsulta.dadosAtendimento.tipoSaida");
                guia.STRindicadorAcidente =
request.getParameter("guiaConsulta.hipoteseDiagnostica.indicadorAcidente");

                guia.STnomeTabela        =
request.getParameter("guiaConsulta.hipoteseDiagnostica.CID.nomeTabela");
                guia.STcodigoDiagnostico     =
request.getParameter("guiaConsulta.hipoteseDiagnostica.CID.codigoDiagnostico");
                guia.STdescricaoDiagnostico  =
request.getParameter("guiaConsulta.hipoteseDiagnostica.CID.descricaoDiagnostico");
                guia.STnomeTabelaSec     =
request.getParameter("guiaConsulta.hipoteseDiagnostica.diagnosticoSecundario.CID.nomeTa
bela");
                guia.STcodigoDiagnosticoSec  =
request.getParameterValues("guiaConsulta.hipoteseDiagnostica.diagnosticoSecundario.CID.c
odigoDiagnostico");
                guia.STdescricaoDiagnosticoSec =
request.getParameter("guiaConsulta.hipoteseDiagnostica.diagnosticoSecundario.CID.descrica
oDiagnostico");
                guia.Stvalor             =
request.getParameter("guiaConsulta.hipoteseDiagnostica.tempoReferidoEvolucaoDoenca.valo
r");
                guia.STunidadeTempo      =
request.getParameter("guiaConsulta.hipoteseDiagnostica.tempoReferidoEvolucaoDoenca.unid
adeTempo");
                guia.STtipoDoenca        =
request.getParameter("guiaConsulta.hipoteseDiagnostica.tipoDoenca");

                guia.STRobservacao = request.getParameter("guiaConsulta.observacao");
```

```
            return guia;
        }

        public static ClGuiaConsulta getGuiaDoMapa(Map<String, String[]> mapa) {
            ClGuiaConsulta guia = new ClGuiaConsulta();

            guia.STRregistroANS        = Util.getPrimeiro(mapa,
"guiaConsulta.identificacaoGuia.registroANS");
            guia.STRdataEmissaoGuia    = Util.getPrimeiro(mapa,
"guiaConsulta.identificacaoGuia.dataEmissaoGuia");
            guia.STRnumeroGuiaPrestador = Util.getPrimeiro(mapa,
"guiaConsulta.identificacaoGuia.numeroGuiaPrestador");
            guia.STRnumeroGuiaOperadora = Util.getPrimeiro(mapa,
"guiaConsulta.identificacaoGuia.numeroGuiaOperadora");

            guia.STRnumeroCarteira          = Util.getPrimeiro(mapa,
"guiaConsulta.beneficiario.numeroCarteira");
            guia.STRnomeBeneficiario        = Util.getPrimeiro(mapa,
"guiaConsulta.beneficiario.nomeBeneficiario");
            guia.STRnomePlano               = Util.getPrimeiro(mapa,
"guiaConsulta.beneficiario.nomePlano");
            guia.STRvalidadeCarteira        = Util.getPrimeiro(mapa,
"guiaConsulta.beneficiario.validadeCarteira");
            guia.STRnumeroCNS               = Util.getPrimeiro(mapa,
"guiaConsulta.beneficiario.numeroCNS");
            guia.STRidentificadorBeneficiario = Util.getPrimeiro(mapa,
"guiaConsulta.beneficiario.identificadorBeneficiario");

            guia.STRCNPJ                    = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.identificacao.CNPJ");
            guia.STRCPF                     = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.identificacao.CPF");
            guia.STRcodigoPrestadorNaOperadora = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperadora");
            guia.STRnomeContratado          = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.nomeContratado");
            guia.STRtipoLogradouro          = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.tipoLogradouro");
            guia.STRlogradouro              = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.logradouro");
            guia.STRnumero                  = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.numero");
            guia.STRcomplemento             = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.complemento");
            guia.STRcodigoIBGEMunicipio     = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.codigoIBGEMunicipio");
            guia.STRmunicipio               = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.municipio");
            guia.STRcodigoUF                = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.codigoUF");
            guia.STRcep                     = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.enderecoContratado.cep");
            guia.STRnumeroCNES              = Util.getPrimeiro(mapa,
"guiaConsulta.dadosContratado.numeroCNES");

            guia.STRnomeProfissional = Util.getPrimeiro(mapa,
"guiaConsulta.profissionalExecutante.nomeProfissional");
            guia.STRsiglaConselho    = Util.getPrimeiro(mapa,
"guiaConsulta.profissionalExecutante.conselhoProfissional.siglaConselho");
```

```
                guia.STRnumeroConselho   = Util.getPrimeiro(mapa,
"guiaConsulta.profissionalExecutante.conselhoProfissional.numeroConselho");
                guia.STRufConselho       = Util.getPrimeiro(mapa,
"guiaConsulta.profissionalExecutante.conselhoProfissional.ufConselho");
                guia.STRcbos             = Util.getPrimeiro(mapa,
"guiaConsulta.profissionalExecutante.cbos");

                // Hipotese Diagnostica
                guia.STRdataAtendimento    = Util.getPrimeiro(mapa,
"guiaConsulta.dadosAtendimento.dataAtendimento");
                guia.STRcodigoTabela       = Util.getPrimeiro(mapa,
"guiaConsulta.dadosAtendimento.procedimento.codigoTabela");
                guia.STRcodigoProcedimento = Util.getPrimeiro(mapa,
"guiaConsulta.dadosAtendimento.procedimento.codigoProcedimento");
                guia.STRtipoConsulta       = Util.getPrimeiro(mapa,
"guiaConsulta.dadosAtendimento.tipoConsulta");
                guia.STRtipoSaida          = Util.getPrimeiro(mapa,
"guiaConsulta.dadosAtendimento.tipoSaida");
                guia.STRindicadorAcidente  = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.indicadorAcidente");

                guia.STnomeTabela          = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.CID.nomeTabela");
                guia.STcodigoDiagnostico   = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.CID.codigoDiagnostico");
                guia.STdescricaoDiagnostico  = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.CID.descricaoDiagnostico");
                guia.STnomeTabelaSec       = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.diagnosticoSecundario.CID.nomeTabela");
                guia.STcodigoDiagnosticoSec  =
mapa.get("guiaConsulta.hipoteseDiagnostica.diagnosticoSecundario.CID.codigoDiagnostico");
                guia.STdescricaoDiagnosticoSec = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.diagnosticoSecundario.CID.descricaoDiagnostico");
                guia.Stvalor               = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.tempoReferidoEvolucaoDoenca.valor");
                guia.STunidadeTempo        = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.tempoReferidoEvolucaoDoenca.unidadeTempo");
                guia.STtipoDoenca          = Util.getPrimeiro(mapa,
"guiaConsulta.hipoteseDiagnostica.tipoDoenca");

                guia.STRobservacao = Util.getPrimeiro(mapa, "guiaConsulta.observacao");

                return guia;
        }

        public String getSTRregistroANS() {
                return STRregistroANS;
        }

        public void setSTRregistroANS(String rregistroANS) {
                STRregistroANS = rregistroANS;
        }

        public String getSTRdataEmissaoGuia() {
                return STRdataEmissaoGuia;
        }

        public void setSTRdataEmissaoGuia(String rdataEmissaoGuia) {
                STRdataEmissaoGuia = rdataEmissaoGuia;
        }
```

```
public String getSTRnumeroGuiaPrestador() {
        return STRnumeroGuiaPrestador;
}

public void setSTRnumeroGuiaPrestador(String rnumeroGuiaPrestador) {
        STRnumeroGuiaPrestador = rnumeroGuiaPrestador;
}

public String getSTRnumeroGuiaOperadora() {
        return STRnumeroGuiaOperadora;
}

public void setSTRnumeroGuiaOperadora(String rnumeroGuiaOperadora) {
        STRnumeroGuiaOperadora = rnumeroGuiaOperadora;
}

public String getSTRnumeroCarteira() {
        return STRnumeroCarteira;
}

public void setSTRnumeroCarteira(String rnumeroCarteira) {
        STRnumeroCarteira = rnumeroCarteira;
}

public String getSTRnomeBeneficiario() {
        return STRnomeBeneficiario;
}

public void setSTRnomeBeneficiario(String rnomeBeneficiario) {
        STRnomeBeneficiario = rnomeBeneficiario;
}

public String getSTRnomePlano() {
        return STRnomePlano;
}

public void setSTRnomePlano(String rnomePlano) {
        STRnomePlano = rnomePlano;
}

public String getSTRvalidadeCarteira() {
        return STRvalidadeCarteira;
}

public void setSTRvalidadeCarteira(String rvalidadeCarteira) {
        STRvalidadeCarteira = rvalidadeCarteira;
}

public String getSTRnumeroCNS() {
        return STRnumeroCNS;
}

public void setSTRnumeroCNS(String rnumeroCNS) {
        STRnumeroCNS = rnumeroCNS;
}

public String getSTRidentificadorBeneficiario() {
        return STRidentificadorBeneficiario;
}
```

```
	public void setSTRidentificadorBeneficiario(String ridentificadorBeneficiario) {
		STRidentificadorBeneficiario = ridentificadorBeneficiario;
	}

	public String getSTRCNPJ() {
		return STRCNPJ;
	}

	public void setSTRCNPJ(String strcnpj) {
		STRCNPJ = strcnpj;
	}

	public String getSTRCPF() {
		return STRCPF;
	}

	public void setSTRCPF(String strcpf) {
		STRCPF = strcpf;
	}

	public String getSTRcodigoPrestadorNaOperadora() {
		return STRcodigoPrestadorNaOperadora;
	}

	public void setSTRcodigoPrestadorNaOperadora(String
rcodigoPrestadorNaOperadora) {
		STRcodigoPrestadorNaOperadora = rcodigoPrestadorNaOperadora;
	}

	public String getSTRnomeContratado() {
		return STRnomeContratado;
	}

	public void setSTRnomeContratado(String rnomeContratado) {
		STRnomeContratado = rnomeContratado;
	}

	public String getSTRtipoLogradouro() {
		return STRtipoLogradouro;
	}

	public void setSTRtipoLogradouro(String rtipoLogradouro) {
		STRtipoLogradouro = rtipoLogradouro;
	}

	public String getSTRlogradouro() {
		return STRlogradouro;
	}

	public void setSTRlogradouro(String rlogradouro) {
		STRlogradouro = rlogradouro;
	}

	public String getSTRnumero() {
		return STRnumero;
	}

	public void setSTRnumero(String rnumero) {
		STRnumero = rnumero;
```

```
}

public String getSTRcomplemento() {
        return STRcomplemento;
}

public void setSTRcomplemento(String rcomplemento) {
        STRcomplemento = rcomplemento;
}

public String getSTRcodigoIBGEMunicipio() {
        return STRcodigoIBGEMunicipio;
}

public void setSTRcodigoIBGEMunicipio(String rcodigoIBGEMunicipio) {
        STRcodigoIBGEMunicipio = rcodigoIBGEMunicipio;
}

public String getSTRmunicipio() {
        return STRmunicipio;
}

public void setSTRmunicipio(String rmunicipio) {
        STRmunicipio = rmunicipio;
}

public String getSTRcodigoUF() {
        return STRcodigoUF;
}

public void setSTRcodigoUF(String rcodigoUF) {
        STRcodigoUF = rcodigoUF;
}

public String getSTRcep() {
        return STRcep;
}

public void setSTRcep(String rcep) {
        STRcep = rcep;
}

public String getSTRnumeroCNES() {
        return STRnumeroCNES;
}

public void setSTRnumeroCNES(String rnumeroCNES) {
        STRnumeroCNES = rnumeroCNES;
}

public String getSTRnomeProfissional() {
        return STRnomeProfissional;
}

public void setSTRnomeProfissional(String rnomeProfissional) {
        STRnomeProfissional = rnomeProfissional;
}

public String getSTRsiglaConselho() {
        return STRsiglaConselho;
```

```
}

public void setSTRsiglaConselho(String rsiglaConselho) {
        STRsiglaConselho = rsiglaConselho;
}

public String getSTRnumeroConselho() {
        return STRnumeroConselho;
}

public void setSTRnumeroConselho(String rnumeroConselho) {
        STRnumeroConselho = rnumeroConselho;
}

public String getSTRufConselho() {
        return STRufConselho;
}

public void setSTRufConselho(String rufConselho) {
        STRufConselho = rufConselho;
}

public String getSTRcbos() {
        return STRcbos;
}

public void setSTRcbos(String rcbos) {
        STRcbos = rcbos;
}

public String getSTRdataAtendimento() {
        return STRdataAtendimento;
}

public void setSTRdataAtendimento(String rdataAtendimento) {
        STRdataAtendimento = rdataAtendimento;
}

public String getSTRcodigoTabela() {
        return STRcodigoTabela;
}

public void setSTRcodigoTabela(String rcodigoTabela) {
        STRcodigoTabela = rcodigoTabela;
}

public String getSTRcodigoProcedimento() {
        return STRcodigoProcedimento;
}

public void setSTRcodigoProcedimento(String rcodigoProcedimento) {
        STRcodigoProcedimento = rcodigoProcedimento;
}

public String getSTRtipoConsulta() {
        return STRtipoConsulta;
}

public void setSTRtipoConsulta(String rtipoConsulta) {
        STRtipoConsulta = rtipoConsulta;
```

```
}

public String getSTRtipoSaida() {
        return STRtipoSaida;
}

public void setSTRtipoSaida(String rtipoSaida) {
        STRtipoSaida = rtipoSaida;
}

public String getSTRindicadorAcidente() {
        return STRindicadorAcidente;
}

public void setSTRindicadorAcidente(String rindicadorAcidente) {
        STRindicadorAcidente = rindicadorAcidente;
}

public String getSTnomeTabela() {
        return STnomeTabela;
}

public void setSTnomeTabela(String tnomeTabela) {
        STnomeTabela = tnomeTabela;
}

public String getSTcodigoDiagnostico() {
        return STcodigoDiagnostico;
}

public void setSTcodigoDiagnostico(String tcodigoDiagnostico) {
        STcodigoDiagnostico = tcodigoDiagnostico;
}

public String getSTdescricaoDiagnostico() {
        return STdescricaoDiagnostico;
}

public void setSTdescricaoDiagnostico(String tdescricaoDiagnostico) {
        STdescricaoDiagnostico = tdescricaoDiagnostico;
}

public String getSTnomeTabelaSec() {
        return STnomeTabelaSec;
}

public void setSTnomeTabelaSec(String tnomeTabelaSec) {
        STnomeTabelaSec = tnomeTabelaSec;
}

public String[] getSTcodigoDiagnosticoSec() {
        return STcodigoDiagnosticoSec;
}

public void setSTcodigoDiagnosticoSec(String[] tcodigoDiagnosticoSec) {
        STcodigoDiagnosticoSec = tcodigoDiagnosticoSec;
}

public String getSTdescricaoDiagnosticoSec() {
        return STdescricaoDiagnosticoSec;
```

```
	}

	public void setSTdescricaoDiagnosticoSec(String tdescricaoDiagnosticoSec) {
		STdescricaoDiagnosticoSec = tdescricaoDiagnosticoSec;
	}

	public String getStvalor() {
		return Stvalor;
	}

	public void setStvalor(String stvalor) {
		Stvalor = stvalor;
	}

	public String getSTunidadeTempo() {
		return STunidadeTempo;
	}

	public void setSTunidadeTempo(String tunidadeTempo) {
		STunidadeTempo = tunidadeTempo;
	}

	public String getSTtipoDoenca() {
		return STtipoDoenca;
	}

	public void setSTtipoDoenca(String ttipoDoenca) {
		STtipoDoenca = ttipoDoenca;
	}

	public String getSTRobservacao() {
		return STRobservacao;
	}

	public void setSTRobservacao(String robservacao) {
		STRobservacao = robservacao;
	}
}
```

Nome do arquivo: CIGuiaHonorarioIndividual.java

```
package com.classes;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import com.util.guia.Util;

public class CIGuiaHonorarioIndividual {
	private String STRregistroANS;
	private String STRdataEmissaoGuia;
	private String STRnumeroGuiaPrestador;
	private String STRnumeroGuiaOperadora;
	private String STRnumeroGuiaPrincipal;
	private String STRnumeroCarteira;
	private String STRnomeBeneficiario;
	private String STRnomePlano;
	private String STRvalidadeCarteira;
```

```
        private String STRnumeroCNS;
        private String STRidentificadorBeneficiario;
        private String STRCNPJ;
        private String STRCPF;
        private String STRcodigoPrestadorNaOperadora;
        private String STRnomeContratado;
        private String STRtipoLogradouro;
        private String STRlogradouro;
        private String STRnumero;
        private String STRcomplemento;
        private String STRcodigoIBGEMunicipio;
        private String STRmunicipio;
        private String STRcodigoUF;
        private String STRcep;
        private String STRnumeroCNES;
        private String STRCNPJExe;
        private String STRCPFExe;
        private String STRcodigoPrestadorNaOperadoraExe;
        private String STRnomeContratadoExe;
        private String STRtipoLogradouroExe;
        private String STRlogradouroExe;
        private String STRnumeroExe;
        private String STRcomplementoExe;
        private String STRcodigoIBGEMunicipioExe;
        private String STRmunicipioExe;
        private String STRcodigoUFExe;
        private String STRcepExe;
        private String STRnumeroCNESExe;
        private String STRnomeExecutanteExe;
        private String STRsiglaConselhoExe;
        private String STRnumeroConselhoExe;
        private String STRufConselhoExe;
        private String STRcodigoCBOSExe;
        private String STRposicaoProfissionalExe;
        private String STRtipoAcomodacaoExe;
        private String[] STRdata;
        private String[] STRhoraInicio;
        private String[] STRhoraFim;
        private String[] STRcodigo;
        private String[] STRtipoTabela;
        private String[] STRdescricao;
        private String[] STRquantidadeRealizada;
        private String[] STRviaAcesso;
        private String[] STRtecnicaUtilizada;
        private String[] STRreducaoAcrescimo;
        private String[] STRvalor;
        private String[] STRvalorTotal;
        private String STRtotalGeralHonorario;
        private String STRobservacao;

        public static CIGuiaHonorarioIndividual getGuiaDoRequest(HttpServletRequest
request){
                CIGuiaHonorarioIndividual guia = new CIGuiaHonorarioIndividual();

                guia.STRregistroANS    =
request.getParameter("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.registroA
NS");
                guia.STRdataEmissaoGuia      =
request.getParameter("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.dataEmis
saoGuia");
```

```
                guia.STRnumeroGuiaPrestador =
request.getParameter("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroG
uiaPrestador");
                guia.STRnumeroGuiaOperadora          =
request.getParameter("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroG
uiaOperadora");
                guia.STRnumeroGuiaPrincipal  =
request.getParameter("guiaHonorarioIndividual.numeroGuiaPrincipal");

                guia.STRnumeroCarteira          =
request.getParameter("guiaHonorarioIndividual.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario        =
request.getParameter("guiaHonorarioIndividual.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano     =
request.getParameter("guiaHonorarioIndividual.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira        =
request.getParameter("guiaHonorarioIndividual.dadosBeneficiario.validadeCarteira");
                guia.STRnumeroCNS   =
request.getParameter("guiaHonorarioIndividual.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario       =
request.getParameter("guiaHonorarioIndividual.dadosBeneficiario.identificadorBeneficiario");

                guia.STRCNPJ =
request.getParameter("guiaHonorarioIndividual.contratado.identificacao.CNPJ");
                guia.STRCPF   =
request.getParameter("guiaHonorarioIndividual.contratado.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadora =
request.getParameter("guiaHonorarioIndividual.contratado.identificacao.codigoPrestadorNaOp
eradora");
                guia.STRnomeContratado          =
request.getParameter("guiaHonorarioIndividual.contratado.nomeContratado");
                guia.STRtipoLogradouro          =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.tipoLogradour
o");
                guia.STRlogradouro      =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.logradouro");
                guia.STRnumero          =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.numero");
                guia.STRcomplemento =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.complemento"
);
                guia.STRcodigoIBGEMunicipio =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.codigoIBGEM
unicipio");
                guia.STRmunicipio       =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.municipio");
                guia.STRcodigoUF        =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.codigoUF");
                guia.STRcep     =
request.getParameter("guiaHonorarioIndividual.contratado.enderecoContratado.cep");
                guia.STRnumeroCNES =
request.getParameter("guiaHonorarioIndividual.contratado.numeroCNES");

                guia.STRCNPJExe         =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacao.CNPJ");
                guia.STRCPFExe          =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadoraExe       =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacao.codigoPres
tadorNaOperadora");
```

```
                guia.STRnomeContratadoExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.nomeContratado");
                guia.STRtipoLogradouroExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.tipo
Logradouro");
                guia.STRlogradouroExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.log
radouro");
                guia.STRnumeroExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.nu
mero");
                guia.STRcomplementoExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.co
mplemento");
                guia.STRcodigoIBGEMunicipioExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.cod
igoIBGEMunicipio");
                guia.STRmunicipioExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.mu
nicipio");
                guia.STRcodigoUFExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.cod
igoUF");
                guia.STRcepExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.cep
");
                guia.STRnumeroCNESExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.numeroCNES");
                guia.STRnomeExecutanteExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
nomeExecutante");
                guia.STRsiglaConselhoExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
conselhoProfissional.numeroConselho");
                guia.STRufConselhoExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
conselhoProfissional.ufConselho");
                guia.STRcodigoCBOSExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
codigoCBOS");
                guia.STRposicaoProfissionalExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.posicaoProfissional");
                guia.STRtipoAcomodacaoExe =
request.getParameter("guiaHonorarioIndividual.contratadoExecutante.tipoAcomodacao");

                guia.STRdata =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.data");
                guia.STRhoraInicio =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.horaInicio");
                guia.STRhoraFim =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.horaFim");
                guia.STRcodigo =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.procedimento.codigo");
```

```
                guia.STRtipoTabela      =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.procedimento.tipoTabela");
                guia.STRdescricao       =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.procedimento.descricao");
                guia.STRquantidadeRealizada  =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.quantidadeRealizada");
                guia.STRviaAcesso       =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.viaAcesso");
                guia.STRtecnicaUtilizada        =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.tecnicaUtilizada");
                guia.STRreducaoAcrescimo        =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.reducaoAcrescimo");
                guia.STRvalor   =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.valor");
                guia.STRvalorTotal      =
request.getParameterValues("guiaHonorarioIndividual.procedimentosExamesRealizados.proce
dimentoRealizado.valorTotal");
                guia.STRtotalGeralHonorario     =
request.getParameter("guiaHonorarioIndividual.procedimentosExamesRealizados.totalGeralHo
norario");
                guia.STRobservacao      =
request.getParameter("guiaHonorarioIndividual.observacao");

                return guia;
        }

        public static ClGuiaHonorarioIndividual getGuiaDoMapa(Map<String, String[]> mapa) {
                ClGuiaHonorarioIndividual guia = new ClGuiaHonorarioIndividual();

                guia.STRregistroANS     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.registroA
NS");
                guia.STRdataEmissaoGuia         =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.dataEmis
saoGuia");
                guia.STRnumeroGuiaPrestador =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroG
uiaPrestador");
                guia.STRnumeroGuiaOperadora             =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroG
uiaOperadora");
                guia.STRnumeroGuiaPrincipal     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.numeroGuiaPrincipal");

                guia.STRnumeroCarteira          =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario        =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira        =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.dadosBeneficiario.validadeCarteira");
```

```
                guia.STRnumeroCNS   =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.dadosBeneficiario.identificadorBeneficiario");

                guia.STRCNPJ =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.identificacao.CNPJ");
                guia.STRCPF   =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadora =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.identificacao.codigoPrestadorNaOp
eradora");
                guia.STRnomeContratado       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.nomeContratado");
                guia.STRtipoLogradouro       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.tipoLogradour
o");
                guia.STRlogradouro     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.logradouro");
                guia.STRnumero       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.numero");
                guia.STRcomplemento =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.complemento"
);
                guia.STRcodigoIBGEMunicipio =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.codigoIBGEM
unicipio");
                guia.STRmunicipio       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.municipio");
                guia.STRcodigoUF       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.codigoUF");
                guia.STRcep    =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.enderecoContratado.cep");
                guia.STRnumeroCNES =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratado.numeroCNES");

                guia.STRCNPJExe       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacao.CNPJ");
                guia.STRCPFExe        =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadoraExe       =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacao.codigoPres
tadorNaOperadora");
                guia.STRnomeContratadoExe   =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.nomeContratado");
                guia.STRtipoLogradouroExe     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.tipo
Logradouro");
                guia.STRlogradouroExe         =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.logr
adouro");
                guia.STRnumeroExe     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.nu
mero");
                guia.STRcomplementoExe        =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.co
mplemento");
                guia.STRcodigoIBGEMunicipioExe        =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.cod
igoIBGEMunicipio");
```

```
guia.STRmunicipioExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.mu
nicipio");
guia.STRcodigoUFExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.cod
igoUF");
guia.STRcepExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.enderecoContratado.cep
");
guia.STRnumeroCNESExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.numeroCNES");
guia.STRnomeExecutanteExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
nomeExecutante");
guia.STRsiglaConselhoExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
conselhoProfissional.siglaConselho");
guia.STRnumeroConselhoExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
conselhoProfissional.numeroConselho");
guia.STRufConselhoExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
conselhoProfissional.ufConselho");
guia.STRcodigoCBOSExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.identificacaoProfissional.
codigoCBOS");
guia.STRposicaoProfissionalExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.posicaoProfissional");
guia.STRtipoAcomodacaoExe =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.contratadoExecutante.tipoAcomodacao");

guia.STRdata =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
data");
guia.STRhoraInicio =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
horaInicio");
guia.STRhoraFim =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
horaFim");
guia.STRcodigo =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
procedimento.codigo");
guia.STRtipoTabela =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
procedimento.tipoTabela");
guia.STRdescricao =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
procedimento.descricao");
guia.STRquantidadeRealizada =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
quantidadeRealizada");
guia.STRviaAcesso =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
viaAcesso");
guia.STRtecnicaUtilizada =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.t
ecnicaUtilizada");
```

```
                guia.STRreducaoAcrescimo     =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.r
educaoAcrescimo");
                guia.STRvalor  =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
valor");
                guia.STRvalorTotal      =
mapa.get("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoRealizado.
valorTotal");
                guia.STRtotalGeralHonorario     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.procedimentosExamesRealizados.totalGeralHo
norario");
                guia.STRobservacao     =
Util.getPrimeiro(mapa,"guiaHonorarioIndividual.observacao");

                return guia;
        }



        public String getSTRregistroANS() {
                return STRregistroANS;
        }
        public void setSTRregistroANS(String rregistroANS) {
                STRregistroANS = rregistroANS;
        }
        public String getSTRdataEmissaoGuia() {
                return STRdataEmissaoGuia;
        }
        public void setSTRdataEmissaoGuia(String rdataEmissaoGuia) {
                STRdataEmissaoGuia = rdataEmissaoGuia;
        }
        public String getSTRnumeroGuiaPrestador() {
                return STRnumeroGuiaPrestador;
        }
        public void setSTRnumeroGuiaPrestador(String rnumeroGuiaPrestador) {
                STRnumeroGuiaPrestador = rnumeroGuiaPrestador;
        }
        public String getSTRnumeroGuiaOperadora() {
                return STRnumeroGuiaOperadora;
        }
        public void setSTRnumeroGuiaOperadora(String rnumeroGuiaOperadora) {
                STRnumeroGuiaOperadora = rnumeroGuiaOperadora;
        }
        public String getSTRnumeroGuiaPrincipal() {
                return STRnumeroGuiaPrincipal;
        }
        public void setSTRnumeroGuiaPrincipal(String rnumeroGuiaPrincipal) {
                STRnumeroGuiaPrincipal = rnumeroGuiaPrincipal;
        }
        public String getSTRnumeroCarteira() {
                return STRnumeroCarteira;
        }
        public void setSTRnumeroCarteira(String rnumeroCarteira) {
                STRnumeroCarteira = rnumeroCarteira;
        }
        public String getSTRnomeBeneficiario() {
                return STRnomeBeneficiario;
        }
```

```
	public void setSTRnomeBeneficiario(String rnomeBeneficiario) {
		STRnomeBeneficiario = rnomeBeneficiario;
	}
	public String getSTRnomePlano() {
		return STRnomePlano;
	}
	public void setSTRnomePlano(String rnomePlano) {
		STRnomePlano = rnomePlano;
	}
	public String getSTRvalidadeCarteira() {
		return STRvalidadeCarteira;
	}
	public void setSTRvalidadeCarteira(String rvalidadeCarteira) {
		STRvalidadeCarteira = rvalidadeCarteira;
	}
	public String getSTRnumeroCNS() {
		return STRnumeroCNS;
	}
	public void setSTRnumeroCNS(String rnumeroCNS) {
		STRnumeroCNS = rnumeroCNS;
	}
	public String getSTRidentificadorBeneficiario() {
		return STRidentificadorBeneficiario;
	}
	public void setSTRidentificadorBeneficiario(String ridentificadorBeneficiario) {
		STRidentificadorBeneficiario = ridentificadorBeneficiario;
	}
	public String getSTRCNPJ() {
		return STRCNPJ;
	}
	public void setSTRCNPJ(String strcnpj) {
		STRCNPJ = strcnpj;
	}
	public String getSTRCPF() {
		return STRCPF;
	}
	public void setSTRCPF(String strcpf) {
		STRCPF = strcpf;
	}
	public String getSTRcodigoPrestadorNaOperadora() {
		return STRcodigoPrestadorNaOperadora;
	}
	public void setSTRcodigoPrestadorNaOperadora(String
rcodigoPrestadorNaOperadora) {
		STRcodigoPrestadorNaOperadora = rcodigoPrestadorNaOperadora;
	}
	public String getSTRnomeContratado() {
		return STRnomeContratado;
	}
	public void setSTRnomeContratado(String rnomeContratado) {
		STRnomeContratado = rnomeContratado;
	}
	public String getSTRtipoLogradouro() {
		return STRtipoLogradouro;
	}
	public void setSTRtipoLogradouro(String rtipoLogradouro) {
		STRtipoLogradouro = rtipoLogradouro;
	}
	public String getSTRlogradouro() {
		return STRlogradouro;
```

```
}
public void setSTRlogradouro(String rlogradouro) {
        STRlogradouro = rlogradouro;
}
public String getSTRnumero() {
        return STRnumero;
}
public void setSTRnumero(String rnumero) {
        STRnumero = rnumero;
}
public String getSTRcomplemento() {
        return STRcomplemento;
}
public void setSTRcomplemento(String rcomplemento) {
        STRcomplemento = rcomplemento;
}
public String getSTRcodigoIBGEMunicipio() {
        return STRcodigoIBGEMunicipio;
}
public void setSTRcodigoIBGEMunicipio(String rcodigoIBGEMunicipio) {
        STRcodigoIBGEMunicipio = rcodigoIBGEMunicipio;
}
public String getSTRmunicipio() {
        return STRmunicipio;
}
public void setSTRmunicipio(String rmunicipio) {
        STRmunicipio = rmunicipio;
}
public String getSTRcodigoUF() {
        return STRcodigoUF;
}
public void setSTRcodigoUF(String rcodigoUF) {
        STRcodigoUF = rcodigoUF;
}
public String getSTRcep() {
        return STRcep;
}
public void setSTRcep(String rcep) {
        STRcep = rcep;
}
public String getSTRnumeroCNES() {
        return STRnumeroCNES;
}
public void setSTRnumeroCNES(String rnumeroCNES) {
        STRnumeroCNES = rnumeroCNES;
}
public String getSTRCNPJExe() {
        return STRCNPJExe;
}
public void setSTRCNPJExe(String exe) {
        STRCNPJExe = exe;
}
public String getSTRCPFExe() {
        return STRCPFExe;
}
public void setSTRCPFExe(String exe) {
        STRCPFExe = exe;
}
public String getSTRcodigoPrestadorNaOperadoraExe() {
        return STRcodigoPrestadorNaOperadoraExe;
```

```
}
public void setSTRcodigoPrestadorNaOperadoraExe(
                String rcodigoPrestadorNaOperadoraExe) {
        STRcodigoPrestadorNaOperadoraExe = rcodigoPrestadorNaOperadoraExe;
}
public String getSTRnomeContratadoExe() {
        return STRnomeContratadoExe;
}
public void setSTRnomeContratadoExe(String rnomeContratadoExe) {
        STRnomeContratadoExe = rnomeContratadoExe;
}
public String getSTRtipoLogradouroExe() {
        return STRtipoLogradouroExe;
}
public void setSTRtipoLogradouroExe(String rtipoLogradouroExe) {
        STRtipoLogradouroExe = rtipoLogradouroExe;
}
public String getSTRlogradouroExe() {
        return STRlogradouroExe;
}
public void setSTRlogradouroExe(String rlogradouroExe) {
        STRlogradouroExe = rlogradouroExe;
}
public String getSTRnumeroExe() {
        return STRnumeroExe;
}
public void setSTRnumeroExe(String rnumeroExe) {
        STRnumeroExe = rnumeroExe;
}
public String getSTRcomplementoExe() {
        return STRcomplementoExe;
}
public void setSTRcomplementoExe(String rcomplementoExe) {
        STRcomplementoExe = rcomplementoExe;
}
public String getSTRcodigoIBGEMunicipioExe() {
        return STRcodigoIBGEMunicipioExe;
}
public void setSTRcodigoIBGEMunicipioExe(String rcodigoIBGEMunicipioExe) {
        STRcodigoIBGEMunicipioExe = rcodigoIBGEMunicipioExe;
}
public String getSTRmunicipioExe() {
        return STRmunicipioExe;
}
public void setSTRmunicipioExe(String rmunicipioExe) {
        STRmunicipioExe = rmunicipioExe;
}
public String getSTRcodigoUFExe() {
        return STRcodigoUFExe;
}
public void setSTRcodigoUFExe(String rcodigoUFExe) {
        STRcodigoUFExe = rcodigoUFExe;
}
public String getSTRcepExe() {
        return STRcepExe;
}
public void setSTRcepExe(String rcepExe) {
        STRcepExe = rcepExe;
}
public String getSTRnumeroCNESExe() {
```

```
        return STRnumeroCNESExe;
}
public void setSTRnumeroCNESExe(String rnumeroCNESExe) {
        STRnumeroCNESExe = rnumeroCNESExe;
}
public String getSTRnomeExecutanteExe() {
        return STRnomeExecutanteExe;
}
public void setSTRnomeExecutanteExe(String rnomeExecutanteExe) {
        STRnomeExecutanteExe = rnomeExecutanteExe;
}
public String getSTRsiglaConselhoExe() {
        return STRsiglaConselhoExe;
}
public void setSTRsiglaConselhoExe(String rsiglaConselhoExe) {
        STRsiglaConselhoExe = rsiglaConselhoExe;
}
public String getSTRnumeroConselhoExe() {
        return STRnumeroConselhoExe;
}
public void setSTRnumeroConselhoExe(String rnumeroConselhoExe) {
        STRnumeroConselhoExe = rnumeroConselhoExe;
}
public String getSTRufConselhoExe() {
        return STRufConselhoExe;
}
public void setSTRufConselhoExe(String rufConselhoExe) {
        STRufConselhoExe = rufConselhoExe;
}
public String getSTRcodigoCBOSExe() {
        return STRcodigoCBOSExe;
}
public void setSTRcodigoCBOSExe(String rcodigoCBOSExe) {
        STRcodigoCBOSExe = rcodigoCBOSExe;
}
public String getSTRposicaoProfissionalExe() {
        return STRposicaoProfissionalExe;
}
public void setSTRposicaoProfissionalExe(String rposicaoProfissionalExe) {
        STRposicaoProfissionalExe = rposicaoProfissionalExe;
}
public String getSTRtipoAcomodacaoExe() {
        return STRtipoAcomodacaoExe;
}
public void setSTRtipoAcomodacaoExe(String rtipoAcomodacaoExe) {
        STRtipoAcomodacaoExe = rtipoAcomodacaoExe;
}
public String[] getSTRdata() {
        return STRdata;
}
public void setSTRdata(String[] rdata) {
        STRdata = rdata;
}
public String[] getSTRhoraInicio() {
        return STRhoraInicio;
}
public void setSTRhoraInicio(String[] rhoraInicio) {
        STRhoraInicio = rhoraInicio;
}
public String[] getSTRhoraFim() {
```

```
        return STRhoraFim;
}
public void setSTRhoraFim(String[] rhoraFim) {
        STRhoraFim = rhoraFim;
}
public String[] getSTRcodigo() {
        return STRcodigo;
}
public void setSTRcodigo(String[] rcodigo) {
        STRcodigo = rcodigo;
}
public String[] getSTRtipoTabela() {
        return STRtipoTabela;
}
public void setSTRtipoTabela(String[] rtipoTabela) {
        STRtipoTabela = rtipoTabela;
}
public String[] getSTRdescricao() {
        return STRdescricao;
}
public void setSTRdescricao(String[] rdescricao) {
        STRdescricao = rdescricao;
}
public String[] getSTRquantidadeRealizada() {
        return STRquantidadeRealizada;
}
public void setSTRquantidadeRealizada(String[] rquantidadeRealizada) {
        STRquantidadeRealizada = rquantidadeRealizada;
}
public String[] getSTRviaAcesso() {
        return STRviaAcesso;
}
public void setSTRviaAcesso(String[] rviaAcesso) {
        STRviaAcesso = rviaAcesso;
}
public String[] getSTRtecnicaUtilizada() {
        return STRtecnicaUtilizada;
}
public void setSTRtecnicaUtilizada(String[] rtecnicaUtilizada) {
        STRtecnicaUtilizada = rtecnicaUtilizada;
}
public String[] getSTRreducaoAcrescimo() {
        return STRreducaoAcrescimo;
}
public void setSTRreducaoAcrescimo(String[] rreducaoAcrescimo) {
        STRreducaoAcrescimo = rreducaoAcrescimo;
}
public String[] getSTRvalor() {
        return STRvalor;
}
public void setSTRvalor(String[] rvalor) {
        STRvalor = rvalor;
}
public String[] getSTRvalorTotal() {
        return STRvalorTotal;
}
public void setSTRvalorTotal(String[] rvalorTotal) {
        STRvalorTotal = rvalorTotal;
}
public String getSTRtotalGeralHonorario() {
```

```
		return STRtotalGeralHonorario;
	}
	public void setSTRtotalGeralHonorario(String rtotalGeralHonorario) {
		STRtotalGeralHonorario = rtotalGeralHonorario;
	}
	public String getSTRobservacao() {
		return STRobservacao;
	}
	public void setSTRobservacao(String robservacao) {
		STRobservacao = robservacao;
	}
}
```

Nome do arquivo: CIGuiaResumoInternacao.java

```
package com.classes;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import com.util.guia.Util;

public class CIGuiaResumoInternacao {
	private String STRregistroANS;
	private String STRdataEmissaoGuia;
	private String STRnumeroGuiaPrestador;
	private String STRnumeroGuiaOperadora;
	private String STRnumeroGuiaPrincipal;
	private String STRnumeroCarteira;
	private String STRnomeBeneficiario;
	private String STRnomePlano;
	private String STRvalidadeCarteira;
	private String STRnumeroCNS;
	private String STRidentificadorBeneficiario;
	private String STRCNPJ;
	private String STRCPF;
	private String STRcodigoPrestadorNaOperadora;
	private String STRnomeContratado;
	private String STRtipoLogradouro;
	private String STRlogradouro;
	private String STRnumero;
	private String STRcomplemento;
	private String STRcodigoIBGEMunicipio;
	private String STRmunicipio;
	private String STRcodigoUF;
	private String STRcep;
	private String STRnumeroCNES;
	private String STRdataAutorizacao;
	private String STRsenhaAutorizacao;
	private String STRvalidadeSenha;
	private String STRcaraterInternacao;
	private String STRacomodacao;
	private String STRdataHoraInternacao;
	private String STRdataHoraSaidaInternacao;
	private String STRtipoInternacao;
	private String STRregimeInternacao;
	private String STRemGestacao;
	private String STRaborto;
```

```
private String STRtranstornoMaternoRelGravidez;
private String STRcomplicacaoPeriodoPuerperio;
private String STRatendimentoRNSalaParto;
private String STRcomplicacaoNeonatal;
private String STRbaixoPeso;
private String STRpartoCesareo;
private String STRpartoNormal;
private String STRnumeroDN;
private String STRqtdNascidosVivosTermo;
private String STRqtdNascidosMortos;
private String STRqtdVivosPrematuros;
private String STRobitoMulher;
private String STRqtdeobitoPrecoce;
private String STRqtdeobitoTardio;
private String STRnomeTabela;
private String STRcodigoDiagnostico;
private String STRdescricaoDiagnostico;
private String STRindicadorAcidente;
private String STRnomeTabelaSec;
private String STRcodigoDiagnosticoSec;
private String STRdescricaoDiagnosticoSec;
private String STRmotivoSaidaInternacaoSec;
private String STRnomeTabelaOb;
private String STRcodigoDiagnosticoOb;
private String STRdescricaoDiagnosticoOb;
private String STRnumeroDeclaracaoOb;
private String[] STRcpfPR;
private String[] STRcodigoPrestadorNaOperadoraPR;
private String STRsiglaConselhoPRCP;
private String STRnumeroConselhoPRCP;
private String STRufConselhoPRCP;
private String[] STRnomeExecutanteID;
private String[] STRsiglaConselhoID;
private String[] STRnumeroConselhoID;
private String[] STRufConselhoID;
private String STRcodigoCBOSID;
private String[] STRcpfID;
private String[] STRposicaoProfissionalID;
private String[] STRcodigoPR;
private String[] STRtipoTabelaPR;
private String[] STRdescricaoPR;
private String[] STRdataPR;
private String[] STRhoraInicioPR;
private String[] STRhoraFimPR;
private String[] STRquantidadeRealizadaPR;
private String[] STRviaAcessoPR;
private String[] STRtecnicaUtilizadaPR;
private String[] STRreducaoAcrescimoPR;
private String[] STRvalorPR;
private String[] STRvalorTotalPR;
private String[] STRcodigoOPM;
private String[] STRtipoTabelaOPM;
private String[] STRdescricaoOPM;
private String[] STRquantidadeOPM;
private String[] STRcodigoBarraOPM;
private String[] STRvalorUnitarioOPM;
private String[] STRvalorTotalOPM;
private String STRvalorTotalOPMOPM;
private String STRcodigoOD;
private String STRtipoTabelaOD;
```

```
        private String STRdescricaoOD;
        private String STRtipoDespesaOD;
        private String STRdataRealizacaoOD;
        private String STRhoraInicialOD;
        private String STRhoraFinalOD;
        private String STRreducaoAcrescimoOD;
        private String STRquantidadeOD;
        private String STRvalorUnitarioOD;
        private String STRvalorTotalOD;
        private String STRtotalGeralOutrasDespesasOD;
        private String STRtipoFaturamento;
        private String STRservicosExecutados;
        private String STRdiarias;
        private String STRtaxas;
        private String STRmateriais;
        private String STRmedicamentos;
        private String STRgases;
        private String STRtotalGeral;
        private String STRobservacao;

        public static CIGuiaResumoInternacao getGuiaDoRequest(HttpServletRequest
request){
                CIGuiaResumoInternacao guia = new CIGuiaResumoInternacao();

                guia.STRregistroANS    =
request.getParameter("guiaResumoInternacao.identificacaoGuiaInternacao.registroANS");
                guia.STRdataEmissaoGuia      =
request.getParameter("guiaResumoInternacao.identificacaoGuiaInternacao.dataEmissaoGuia"
);
                guia.STRnumeroGuiaPrestador =
request.getParameter("guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaPresta
dor");
                guia.STRnumeroGuiaOperadora          =
request.getParameter("guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaOpera
dora");
                guia.STRnumeroGuiaPrincipal  =
request.getParameter("guiaResumoInternacao.numeroGuiaSolicitacao");

                guia.STRnumeroCarteira         =
request.getParameter("guiaResumoInternacao.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario       =
request.getParameter("guiaResumoInternacao.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano    =
request.getParameter("guiaResumoInternacao.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira       =
request.getParameter("guiaResumoInternacao.dadosBeneficiario.validadeCarteira");
                guia.STRnumeroCNS    =
request.getParameter("guiaResumoInternacao.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario        =
request.getParameter("guiaResumoInternacao.dadosBeneficiario.identificadorBeneficiario");

                guia.STRCNPJ =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.identificacao.CNPJ");
                guia.STRCPF    =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadora =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.identificacao.codigoPre
stadorNaOperadora");
                guia.STRnomeContratado       =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.nomeContratado");
```

```
                guia.STRtipoLogradouro          =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.ti
poLogradouro");
                guia.STRlogradouro      =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.lo
gradouro");
                guia.STRnumero          =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.n
umero");
                guia.STRcomplemento =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.c
omplemento");
                guia.STRcodigoIBGEMunicipio =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.c
odigoIBGEMunicipio");
                guia.STRmunicipio       =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.m
unicipio");
                guia.STRcodigoUF        =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.c
odigoUF");
                guia.STRcep     =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.c
ep");
                guia.STRnumeroCNES =
request.getParameter("guiaResumoInternacao.identificacaoExecutante.numeroCNES");

                guia.STRdataAutorizacao         =
request.getParameter("guiaResumoInternacao.dadosAutorizacao.dataAutorizacao");
                guia.STRsenhaAutorizacao        =
request.getParameter("guiaResumoInternacao.dadosAutorizacao.senhaAutorizacao");
                guia.STRvalidadeSenha           =
request.getParameter("guiaResumoInternacao.dadosAutorizacao.validadeSenha");

                guia.STRcaraterInternacao       =
request.getParameter("guiaResumoInternacao.caraterInternacao");

                guia.STRacomodacao =
request.getParameter("guiaResumoInternacao.acomodacao");

                guia.STRdataHoraInternacao      =
request.getParameter("guiaResumoInternacao.dataHoraInternacao");
                guia.STRdataHoraSaidaInternacao         =
request.getParameter("guiaResumoInternacao.dataHoraSaidaInternacao");
                guia.STRtipoInternacao=
request.getParameter("guiaResumoInternacao.tipoInternacao");
                guia.STRregimeInternacao        =
request.getParameter("guiaResumoInternacao.regimeInternacao");

                guia.STRemGestacao =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.emGestacao");
                guia.STRaborto=
request.getParameter("guiaResumoInternacao.internacaoObstetrica.aborto");
                guia.STRtranstornoMaternoRelGravidez            =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.transtornoMaternoRelGra
videz");
                guia.STRcomplicacaoPeriodoPuerperio =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.complicacaoPeriodoPuerp
erio");
```

```
                guia.STRatendimentoRNSalaParto      =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.atendimentoRNSalaParto
");
                guia.STRcomplicacaoNeonatal =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.complicacaoNeonatal");
                guia.STRbaixoPeso     =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.baixoPeso");
                guia.STRpartoCesareo =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.partoCesareo");
                guia.STRpartoNormal   =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.partoNormal");

                guia.STRnumeroDN     =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.declaracoesNascidosVivo
s.numeroDN");
                guia.STRqtdNascidosVivosTermo       =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.qtdNascidosVivosTermo")
;
                guia.STRqtdNascidosMortos     =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.qtdNascidosMortos");
                guia.STRqtdVivosPrematuros   =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.qtdVivosPrematuros");
                guia.STRobitoMulher    =
request.getParameter("guiaResumoInternacao.internacaoObstetrica.obitoMulher");

                guia.STRqtdeobitoPrecoce      =
request.getParameter("guiaResumoInternacao.obitoNeonatal.qtdeobitoPrecoce");
                guia.STRqtdeobitoTardio       =
request.getParameter("guiaResumoInternacao.obitoNeonatal.qtdeobitoTardio");

                guia.STRnomeTabela   =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticoPrinci
pal.nomeTabela");
                guia.STRcodigoDiagnostico     =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticoPrinci
pal.codigoDiagnostico");
                guia.STRdescricaoDiagnostico =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticoPrinci
pal.descricaoDiagnostico");
                guia.STRindicadorAcidente     =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.indicadorAcidente
");

                guia.STRnomeTabelaSec       =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticosSecu
ndarios.CID.nomeTabela");
                guia.STRcodigoDiagnosticoSec=
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticosSecu
ndarios.CID.codigoDiagnostico");
                guia.STRdescricaoDiagnosticoSec     =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticosSecu
ndarios.CID.descricaoDiagnostico");
                guia.STRmotivoSaidaInternacaoSec    =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.motivoSaidaIntern
acao");

                guia.STRnomeTabelaOb        =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.obito.CID.nomeT
abela");
```

```
guia.STRcodigoDiagnosticoOb =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.obito.CID.codigo
Diagnostico");
guia.STRdescricaoDiagnosticoOb =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.obito.CID.descric
aoDiagnostico");
guia.STRnumeroDeclaracaoOb =
request.getParameter("guiaResumoInternacao.diagnosticosSaidaInternacao.obito.numeroDecl
aracao");

guia.STRcpfPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.codigoProfissional.cpf");
guia.STRcodigoPrestadorNaOperadoraPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.codigoProfissional.codigoPrestadorNaOperadora");

guia.STRsiglaConselhoPRCP =
request.getParameter("guiaResumoInternacao.procedimentosRealizados.procedimentos.equip
e.membroEquipe.codigoProfissional.conselhoProfissional.siglaConselho");
guia.STRnumeroConselhoPRCP =
request.getParameter("guiaResumoInternacao.procedimentosRealizados.procedimentos.equip
e.membroEquipe.codigoProfissional.conselhoProfissional.numeroConselho");
guia.STRufConselhoPRCP =
request.getParameter("guiaResumoInternacao.procedimentosRealizados.procedimentos.equip
e.membroEquipe.codigoProfissional.conselhoProfissional.ufConselho");

guia.STRnomeExecutanteID =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.identificacaoProfissional.nomeExecutante");
guia.STRsiglaConselhoID =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.identificacaoProfissional.conselhoProfissional.siglaConselho");
guia.STRnumeroConselhoID =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.identificacaoProfissional.conselhoProfissional.numeroConselho");
guia.STRufConselhoID =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.identificacaoProfissional.conselhoProfissional.ufConselho");
guia.STRcodigoCBOSID =
request.getParameter("guiaResumoInternacao.procedimentosRealizados.procedimentos.equip
e.membroEquipe.identificacaoProfissional.codigoCBOS");
guia.STRcpfID =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.cpf");
guia.STRposicaoProfissionalID =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.equipe.membroEquipe.posicaoProfissional");

guia.STRcodigoPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.procedimento.codigo");
guia.STRtipoTabelaPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.procedimento.tipoTabela");
guia.STRdescricaoPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.procedimento.descricao");
```

```
                guia.STRdataPR          =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.data");
                guia.STRhoraInicioPR    =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.horaInicio");
                guia.STRhoraFimPR       =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.horaFim");
                guia.STRquantidadeRealizadaPR       =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.quantidadeRealizada");
                guia.STRviaAcessoPR     =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.viaAcesso");
                guia.STRtecnicaUtilizadaPR      =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.tecnicaUtilizada");
                guia.STRreducaoAcrescimoPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.reducaoAcrescimo");
                guia.STRvalorPR         =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.valor"); .
                guia.STRvalorTotalPR =
request.getParameterValues("guiaResumoInternacao.procedimentosRealizados.procedimento
s.valorTotal");

                guia.STRcodigoOPM       =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.
OPM.codigo");
                guia.STRtipoTabelaOPM           =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.
OPM.tipoTabela");
                guia.STRdescricaoOPM            =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.
OPM.descricao");
                guia.STRquantidadeOPM           =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.q
uantidade");
                guia.STRcodigoBarraOPM          =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.c
odigoBarra");
                guia.STRvalorUnitarioOPM        =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.v
alorUnitario");
                guia.STRvalorTotalOPM           =
request.getParameterValues("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.v
alorTotal");
                guia.STRvalorTotalOPMOPM        =
request.getParameter("guiaResumoInternacao.OPMUtilizadas.valorTotalOPM");

                guia.STRcodigoOD        =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.identificadorDespesa.
codigo");
                guia.STRtipoTabelaOD =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.identificadorDespesa.
tipoTabela");
```

```
                guia.STRdescricaoOD =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.identificadorDespesa.
descricao");
                guia.STRtipoDespesaOD       =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.tipoDespesa");
                guia.STRdataRealizacaoOD    =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.dataRealizacao");
                guia.STRhoraInicialOD =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.horaInicial");
                guia.STRhoraFinalOD  =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.horaFinal");
                guia.STRreducaoAcrescimoOD =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.reducaoAcrescimo");
                guia.STRquantidadeOD=
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.quantidade");
                guia.STRvalorUnitarioOD      =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.valorUnitario");
                guia.STRvalorTotalOD  =
request.getParameter("guiaResumoInternacao.outrasDespesas.despesa.valorTotal");
                guia.STRtotalGeralOutrasDespesasOD =
request.getParameter("guiaResumoInternacao.outrasDespesas.totalGeralOutrasDespesas");

                guia.STRtipoFaturamento      =
request.getParameter("guiaResumoInternacao.tipoFaturamento");
                guia.STRservicosExecutados   =
request.getParameter("guiaResumoInternacao.valorTotal.servicosExecutados");
                guia.STRdiarias          =
request.getParameter("guiaResumoInternacao.valorTotal.diarias");
                guia.STRtaxas =
request.getParameter("guiaResumoInternacao.valorTotal.taxas");
                guia.STRmateriais        =
request.getParameter("guiaResumoInternacao.valorTotal.materiais");
                guia.STRmedicamentos         =
request.getParameter("guiaResumoInternacao.valorTotal.medicamentos");
                guia.STRgases =
request.getParameter("guiaResumoInternacao.valorTotal.gases");
                guia.STRtotalGeral       =
request.getParameter("guiaResumoInternacao.valorTotal.totalGeral");

                guia.STRobservacao    =
request.getParameter("guiaResumoInternacao.observacao");

                return guia;
        }

        public static ClGuiaResumoInternacao getGuiaDoMapa(Map<String, String[]> mapa) {
                ClGuiaResumoInternacao guia = new ClGuiaResumoInternacao();

                guia.STRregistroANS    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoGuiaInternacao.registroANS");
                guia.STRdataEmissaoGuia       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoGuiaInternacao.dataEmissaoGuia");
                guia.STRnumeroGuiaPrestador = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaPrestador");
                guia.STRnumeroGuiaOperadora          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaOperadora");
                guia.STRnumeroGuiaPrincipal  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.numeroGuiaSolicitacao");
```

```
                guia.STRnumeroCarteira          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosBeneficiario.validadeCarteira");
                guia.STRnumeroCNS    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosBeneficiario.identificadorBeneficiario");

                guia.STRCNPJ = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.identificacao.CNPJ");
                guia.STRCPF    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadora = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.identificacao.codigoPrestadorNaOperadora");
                guia.STRnomeContratado          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.nomeContratado");
                guia.STRtipoLogradouro          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.tipoLogradouro");
                guia.STRlogradouro      = Util.getPrimeiro(mapa,        .
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.logradouro");
                guia.STRnumero          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.numero");
                guia.STRcomplemento  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.complemento");
                guia.STRcodigoIBGEMunicipio  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.codigoIBGEMunicipio");
                guia.STRmunicipio       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.municipio");
                guia.STRcodigoUF        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.codigoUF");
                guia.STRcep     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.enderecoContratado.cep");
                guia.STRnumeroCNES = Util.getPrimeiro(mapa,
"guiaResumoInternacao.identificacaoExecutante.numeroCNES");

                guia.STRdataAutorizacao         = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosAutorizacao.dataAutorizacao");
                guia.STRsenhaAutorizacao        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosAutorizacao.senhaAutorizacao");
                guia.STRvalidadeSenha           = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dadosAutorizacao.validadeSenha");

                guia.STRcaraterInternacao       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.caraterInternacao");

                guia.STRacomodacao   = Util.getPrimeiro(mapa,
"guiaResumoInternacao.acomodacao");

                guia.STRdataHoraInternacao    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dataHoraInternacao");
                guia.STRdataHoraSaidaInternacao        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.dataHoraSaidaInternacao");
                guia.STRtipoInternacao = Util.getPrimeiro(mapa,
"guiaResumoInternacao.tipoInternacao");
```

```
                guia.STRregimeInternacao      = Util.getPrimeiro(mapa,
"guiaResumoInternacao.regimeInternacao");

                guia.STRemGestacao  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.emGestacao");
                guia.STRaborto= Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.aborto");
                guia.STRtranstornoMaternoRelGravidez          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.transtornoMaternoRelGravidez");
                guia.STRcomplicacaoPeriodoPuerperio = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.complicacaoPeriodoPuerperio");
                guia.STRatendimentoRNSalaParto      = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.atendimentoRNSalaParto");
                guia.STRcomplicacaoNeonatal = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.complicacaoNeonatal");
                guia.STRbaixoPeso     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.baixoPeso");
                guia.STRpartoCesareo = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.partoCesareo");
                guia.STRpartoNormal  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.partoNormal");

                guia.STRnumeroDN     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.declaracoesNascidosVivos.numeroDN");
                guia.STRqtdNascidosVivosTermo       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.qtdNascidosVivosTermo");
                guia.STRqtdNascidosMortos    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.qtdNascidosMortos");
                guia.STRqtdVivosPrematuros   = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.qtdVivosPrematuros");
                guia.STRobitoMulher    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.internacaoObstetrica.obitoMulher");

                guia.STRqtdeobitoPrecoce      = Util.getPrimeiro(mapa,
"guiaResumoInternacao.obitoNeonatal.qtdeobitoPrecoce");
                guia.STRqtdeobitoTardio       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.obitoNeonatal.qtdeobitoTardio");

                guia.STRnomeTabela  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticoPrincipal.nomeTabela");
                guia.STRcodigoDiagnostico     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticoPrincipal.codigoDiagnostico")
;
                guia.STRdescricaoDiagnostico = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticoPrincipal.descricaoDiagnosti
co");
                guia.STRindicadorAcidente     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.indicadorAcidente");

                guia.STRnomeTabelaSec       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticosSecundarios.CID.nomeTab
ela");
                guia.STRcodigoDiagnosticoSec= Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticosSecundarios.CID.codigoDia
gnostico");
                guia.STRdescricaoDiagnosticoSec      = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.diagnosticosSecundarios.CID.descricao
Diagnostico");
                guia.STRmotivoSaidaInternacaoSec     = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.motivoSaidaInternacao");
```

```
                guia.STRnomeTabelaOb          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.obito.CID.nomeTabela");
                guia.STRcodigoDiagnosticoOb  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.obito.CID.codigoDiagnostico");
                guia.STRdescricaoDiagnosticoOb        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.obito.CID.descricaoDiagnostico");
                guia.STRnumeroDeclaracaoOb = Util.getPrimeiro(mapa,
"guiaResumoInternacao.diagnosticosSaidaInternacao.obito.numeroDeclaracao");

                guia.STRcpfPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.codigoProfissional.cpf");
                guia.STRcodigoPrestadorNaOperadoraPR      =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.codigoProfissional.codigoPrestadorNaOperadora");

                guia.STRsiglaConselhoPRCP   = Util.getPrimeiro(mapa,
"guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroEquipe.codi
goProfissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoPRCP          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroEquipe.codi
goProfissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoPRCP      = Util.getPrimeiro(mapa,
"guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroEquipe.codi
goProfissional.conselhoProfissional.ufConselho");

                guia.STRnomeExecutanteID    =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.identificacaoProfissional.nomeExecutante");
                guia.STRsiglaConselhoID       =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.identificacaoProfissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoID    =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.identificacaoProfissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoID =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.identificacaoProfissional.conselhoProfissional.ufConselho");
                guia.STRcodigoCBOSID          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroEquipe.ident
ificacaoProfissional.codigoCBOS");
                guia.STRcpfID =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.cpf");
                guia.STRposicaoProfissionalID =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.equipe.membroE
quipe.posicaoProfissional");

                guia.STRcodigoPR      =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.procedimento.cod
igo");
                guia.STRtipoTabelaPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.procedimento.tipo
Tabela");
                guia.STRdescricaoPR  =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.procedimento.des
cricao");
                guia.STRdataPR        =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.data");
```

```
                guia.STRhoraInicioPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.horaInicio");
                guia.STRhoraFimPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.horaFim");
                guia.STRquantidadeRealizadaPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.quantidadeRealiz
ada");
                guia.STRviaAcessoPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.viaAcesso");
                guia.STRtecnicaUtilizadaPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.tecnicaUtilizada");
                guia.STRreducaoAcrescimoPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.reducaoAcrescim
o");
                guia.STRvalorPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.valor");
                guia.STRvalorTotalPR =
mapa.get("guiaResumoInternacao.procedimentosRealizados.procedimentos.valorTotal");

                guia.STRcodigoOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.OPM.codigo");
                guia.STRtipoTabelaOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.OPM.tipoTabela");
                guia.STRdescricaoOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.OPM.descricao");
                guia.STRquantidadeOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.quantidade");
                guia.STRcodigoBarraOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.codigoBarra");
                guia.STRvalorUnitarioOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.valorUnitario");
                guia.STRvalorTotalOPM =
mapa.get("guiaResumoInternacao.OPMUtilizadas.OPM.identificacaoOPM.valorTotal");
                guia.STRvalorTotalOPMOPM = Util.getPrimeiro(mapa,
"guiaResumoInternacao.OPMUtilizadas.valorTotalOPM");

                guia.STRcodigoOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.identificadorDespesa.codigo");
                guia.STRtipoTabelaOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.identificadorDespesa.tipoTabela");
                guia.STRdescricaoOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.identificadorDespesa.descricao");
                guia.STRtipoDespesaOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.tipoDespesa");
                guia.STRdataRealizacaoOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.dataRealizacao");
                guia.STRhoraInicialOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.horaInicial");
                guia.STRhoraFinalOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.horaFinal");
                guia.STRreducaoAcrescimoOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.reducaoAcrescimo");
                guia.STRquantidadeOD= Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.quantidade");
                guia.STRvalorUnitarioOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.valorUnitario");
                guia.STRvalorTotalOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.despesa.valorTotal");
                guia.STRtotalGeralOutrasDespesasOD = Util.getPrimeiro(mapa,
"guiaResumoInternacao.outrasDespesas.totalGeralOutrasDespesas");
```

```
                guia.STRtipoFaturamento        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.tipoFaturamento");
                guia.STRservicosExecutados    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.servicosExecutados");
                guia.STRdiarias          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.diarias");
                guia.STRtaxas  = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.taxas");
                guia.STRmateriais        = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.materiais");
                guia.STRmedicamentos          = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.medicamentos");
                guia.STRgases = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.gases");
                guia.STRtotalGeral       = Util.getPrimeiro(mapa,
"guiaResumoInternacao.valorTotal.totalGeral");

                guia.STRobservacao    = Util.getPrimeiro(mapa,
"guiaResumoInternacao.observacao");

                return guia;
        }

        public String getSTRregistroANS() {
                return STRregistroANS;
        }
        public void setSTRregistroANS(String rregistroANS) {
                STRregistroANS = rregistroANS;
        }
        public String getSTRdataEmissaoGuia() {
                return STRdataEmissaoGuia;
        }
        public void setSTRdataEmissaoGuia(String rdataEmissaoGuia) {
                STRdataEmissaoGuia = rdataEmissaoGuia;
        }
        public String getSTRnumeroGuiaPrestador() {
                return STRnumeroGuiaPrestador;
        }
        public void setSTRnumeroGuiaPrestador(String rnumeroGuiaPrestador) {
                STRnumeroGuiaPrestador = rnumeroGuiaPrestador;
        }
        public String getSTRnumeroGuiaOperadora() {
                return STRnumeroGuiaOperadora;
        }
        public void setSTRnumeroGuiaOperadora(String rnumeroGuiaOperadora) {
                STRnumeroGuiaOperadora = rnumeroGuiaOperadora;
        }
        public String getSTRnumeroGuiaPrincipal() {
                return STRnumeroGuiaPrincipal;
        }
        public void setSTRnumeroGuiaPrincipal(String rnumeroGuiaPrincipal) {
                STRnumeroGuiaPrincipal = rnumeroGuiaPrincipal;
        }
        public String getSTRnumeroCarteira() {
                return STRnumeroCarteira;
        }
        public void setSTRnumeroCarteira(String rnumeroCarteira) {
                STRnumeroCarteira = rnumeroCarteira;
        }
```

```
	public String getSTRnomeBeneficiario() {
		return STRnomeBeneficiario;
	}
	public void setSTRnomeBeneficiario(String rnomeBeneficiario) {
		STRnomeBeneficiario = rnomeBeneficiario;
	}
	public String getSTRnomePlano() {
		return STRnomePlano;
	}
	public void setSTRnomePlano(String rnomePlano) {
		STRnomePlano = rnomePlano;
	}
	public String getSTRvalidadeCarteira() {
		return STRvalidadeCarteira;
	}
	public void setSTRvalidadeCarteira(String rvalidadeCarteira) {
		STRvalidadeCarteira = rvalidadeCarteira;
	}
	public String getSTRnumeroCNS() {
		return STRnumeroCNS;
	}
	public void setSTRnumeroCNS(String rnumeroCNS) {
		STRnumeroCNS = rnumeroCNS;
	}
	public String getSTRidentificadorBeneficiario() {
		return STRidentificadorBeneficiario;
	}
	public void setSTRidentificadorBeneficiario(String ridentificadorBeneficiario) {
		STRidentificadorBeneficiario = ridentificadorBeneficiario;
	}
	public String getSTRCNPJ() {
		return STRCNPJ;
	}
	public void setSTRCNPJ(String strcnpj) {
		STRCNPJ = strcnpj;
	}
	public String getSTRCPF() {
		return STRCPF;
	}
	public void setSTRCPF(String strcpf) {
		STRCPF = strcpf;
	}
	public String getSTRcodigoPrestadorNaOperadora() {
		return STRcodigoPrestadorNaOperadora;
	}
	public void setSTRcodigoPrestadorNaOperadora(String
rcodigoPrestadorNaOperadora) {
		STRcodigoPrestadorNaOperadora = rcodigoPrestadorNaOperadora;
	}
	public String getSTRnomeContratado() {
		return STRnomeContratado;
	}
	public void setSTRnomeContratado(String rnomeContratado) {
		STRnomeContratado = rnomeContratado;
	}
	public String getSTRtipoLogradouro() {
		return STRtipoLogradouro;
	}
	public void setSTRtipoLogradouro(String rtipoLogradouro) {
		STRtipoLogradouro = rtipoLogradouro;
```

```
}
public String getSTRlogradouro() {
        return STRlogradouro;
}
public void setSTRlogradouro(String rlogradouro) {
        STRlogradouro = rlogradouro;
}
public String getSTRnumero() {
        return STRnumero;
}
public void setSTRnumero(String rnumero) {
        STRnumero = rnumero;
}
public String getSTRcomplemento() {
        return STRcomplemento;
}
public void setSTRcomplemento(String rcomplemento) {
        STRcomplemento = rcomplemento;
}
public String getSTRcodigoIBGEMunicipio() {
        return STRcodigoIBGEMunicipio;
}
public void setSTRcodigoIBGEMunicipio(String rcodigoIBGEMunicipio) {
        STRcodigoIBGEMunicipio = rcodigoIBGEMunicipio;
}
public String getSTRmunicipio() {
        return STRmunicipio;
}
public void setSTRmunicipio(String rmunicipio) {
        STRmunicipio = rmunicipio;
}
public String getSTRcodigoUF() {
        return STRcodigoUF;
}
public void setSTRcodigoUF(String rcodigoUF) {
        STRcodigoUF = rcodigoUF;
}
public String getSTRcep() {
        return STRcep;
}
public void setSTRcep(String rcep) {
        STRcep = rcep;
}
public String getSTRnumeroCNES() {
        return STRnumeroCNES;
}
public void setSTRnumeroCNES(String rnumeroCNES) {
        STRnumeroCNES = rnumeroCNES;
}
public String getSTRdataAutorizacao() {
        return STRdataAutorizacao;
}
public void setSTRdataAutorizacao(String rdataAutorizacao) {
        STRdataAutorizacao = rdataAutorizacao;
}
public String getSTRsenhaAutorizacao() {
        return STRsenhaAutorizacao;
}
public void setSTRsenhaAutorizacao(String rsenhaAutorizacao) {
        STRsenhaAutorizacao = rsenhaAutorizacao;
```

```
}
public String getSTRvalidadeSenha() {
        return STRvalidadeSenha;
}
public void setSTRvalidadeSenha(String rvalidadeSenha) {
        STRvalidadeSenha = rvalidadeSenha;
}
public String getSTRcaraterInternacao() {
        return STRcaraterInternacao;
}
public void setSTRcaraterInternacao(String rcaraterInternacao) {
        STRcaraterInternacao = rcaraterInternacao;
}
public String getSTRacomodacao() {
        return STRacomodacao;
}
public void setSTRacomodacao(String racomodacao) {
        STRacomodacao = racomodacao;
}
public String getSTRdataHoraInternacao() {
        return STRdataHoraInternacao;
}
public void setSTRdataHoraInternacao(String rdataHoraInternacao) {
        STRdataHoraInternacao = rdataHoraInternacao;
}
public String getSTRdataHoraSaidaInternacao() {
        return STRdataHoraSaidaInternacao;
}
public void setSTRdataHoraSaidaInternacao(String rdataHoraSaidaInternacao) {
        STRdataHoraSaidaInternacao = rdataHoraSaidaInternacao;
}
public String getSTRtipoInternacao() {
        return STRtipoInternacao;
}
public void setSTRtipoInternacao(String rtipoInternacao) {
        STRtipoInternacao = rtipoInternacao;
}
public String getSTRregimeInternacao() {
        return STRregimeInternacao;
}
public void setSTRregimeInternacao(String rregimeInternacao) {
        STRregimeInternacao = rregimeInternacao;
}
public String getSTRemGestacao() {
        return STRemGestacao;
}
public void setSTRemGestacao(String remGestacao) {
        STRemGestacao = remGestacao;
}
public String getSTRaborto() {
        return STRaborto;
}
public void setSTRaborto(String raborto) {
        STRaborto = raborto;
}
public String getSTRtranstornoMaternoRelGravidez() {
        return STRtranstornoMaternoRelGravidez;
}
public void setSTRtranstornoMaternoRelGravidez(
                String rtranstornoMaternoRelGravidez) {
```

```
        STRtranstornoMaternoRelGravidez = rtranstornoMaternoRelGravidez;
}
public String getSTRcomplicacaoPeriodoPuerperio() {
        return STRcomplicacaoPeriodoPuerperio;
}
public void setSTRcomplicacaoPeriodoPuerperio(
                String rcomplicacaoPeriodoPuerperio) {
        STRcomplicacaoPeriodoPuerperio = rcomplicacaoPeriodoPuerperio;
}
public String getSTRatendimentoRNSalaParto() {
        return STRatendimentoRNSalaParto;
}
public void setSTRatendimentoRNSalaParto(String ratendimentoRNSalaParto) {
        STRatendimentoRNSalaParto = ratendimentoRNSalaParto;
}
public String getSTRcomplicacaoNeonatal() {
        return STRcomplicacaoNeonatal;
}
public void setSTRcomplicacaoNeonatal(String rcomplicacaoNeonatal) {
        STRcomplicacaoNeonatal = rcomplicacaoNeonatal;
}
public String getSTRbaixoPeso() {
        return STRbaixoPeso;
}
public void setSTRbaixoPeso(String rbaixoPeso) {
        STRbaixoPeso = rbaixoPeso;
}
public String getSTRpartoCesareo() {
        return STRpartoCesareo;
}
public void setSTRpartoCesareo(String rpartoCesareo) {
        STRpartoCesareo = rpartoCesareo;
}
public String getSTRpartoNormal() {
        return STRpartoNormal;
}
public void setSTRpartoNormal(String rpartoNormal) {
        STRpartoNormal = rpartoNormal;
}
public String getSTRnumeroDN() {
        return STRnumeroDN;
}
public void setSTRnumeroDN(String rnumeroDN) {
        STRnumeroDN = rnumeroDN;
}
public String getSTRqtdNascidosVivosTermo() {
        return STRqtdNascidosVivosTermo;
}
public void setSTRqtdNascidosVivosTermo(String rqtdNascidosVivosTermo) {
        STRqtdNascidosVivosTermo = rqtdNascidosVivosTermo;
}
public String getSTRqtdNascidosMortos() {
        return STRqtdNascidosMortos;
}
public void setSTRqtdNascidosMortos(String rqtdNascidosMortos) {
        STRqtdNascidosMortos = rqtdNascidosMortos;
}
public String getSTRqtdVivosPrematuros() {
        return STRqtdVivosPrematuros;
}
```

```
public void setSTRqtdVivosPrematuros(String rqtdVivosPrematuros) {
        STRqtdVivosPrematuros = rqtdVivosPrematuros;
}
public String getSTRobitoMulher() {
        return STRobitoMulher;
}
public void setSTRobitoMulher(String robitoMulher) {
        STRobitoMulher = robitoMulher;
}
public String getSTRqtdeobitoPrecoce() {
        return STRqtdeobitoPrecoce;
}
public void setSTRqtdeobitoPrecoce(String rqtdeobitoPrecoce) {
        STRqtdeobitoPrecoce = rqtdeobitoPrecoce;
}
public String getSTRqtdeobitoTardio() {
        return STRqtdeobitoTardio;
}
public void setSTRqtdeobitoTardio(String rqtdeobitoTardio) {
        STRqtdeobitoTardio = rqtdeobitoTardio;
}
public String getSTRnomeTabela() {
        return STRnomeTabela;
}
public void setSTRnomeTabela(String rnomeTabela) {
        STRnomeTabela = rnomeTabela;
}
public String getSTRcodigoDiagnostico() {
        return STRcodigoDiagnostico;
}
public void setSTRcodigoDiagnostico(String rcodigoDiagnostico) {
        STRcodigoDiagnostico = rcodigoDiagnostico;
}
public String getSTRdescricaoDiagnostico() {
        return STRdescricaoDiagnostico;
}
public void setSTRdescricaoDiagnostico(String rdescricaoDiagnostico) {
        STRdescricaoDiagnostico = rdescricaoDiagnostico;
}
public String getSTRindicadorAcidente() {
        return STRindicadorAcidente;
}
public void setSTRindicadorAcidente(String rindicadorAcidente) {
        STRindicadorAcidente = rindicadorAcidente;
}
public String getSTRnomeTabelaSec() {
        return STRnomeTabelaSec;
}
public void setSTRnomeTabelaSec(String rnomeTabelaSec) {
        STRnomeTabelaSec = rnomeTabelaSec;
}
public String getSTRcodigoDiagnosticoSec() {
        return STRcodigoDiagnosticoSec;
}
public void setSTRcodigoDiagnosticoSec(String rcodigoDiagnosticoSec) {
        STRcodigoDiagnosticoSec = rcodigoDiagnosticoSec;
}
public String getSTRdescricaoDiagnosticoSec() {
        return STRdescricaoDiagnosticoSec;
}
```

```
public void setSTRdescricaoDiagnosticoSec(String rdescricaoDiagnosticoSec) {
        STRdescricaoDiagnosticoSec = rdescricaoDiagnosticoSec;
}
public String getSTRmotivoSaidaInternacaoSec() {
        return STRmotivoSaidaInternacaoSec;
}
public void setSTRmotivoSaidaInternacaoSec(String rmotivoSaidaInternacaoSec) {
        STRmotivoSaidaInternacaoSec = rmotivoSaidaInternacaoSec;
}
public String getSTRnomeTabelaOb() {
        return STRnomeTabelaOb;
}
public void setSTRnomeTabelaOb(String rnomeTabelaOb) {
        STRnomeTabelaOb = rnomeTabelaOb;
}
public String getSTRcodigoDiagnosticoOb() {
        return STRcodigoDiagnosticoOb;
}
public void setSTRcodigoDiagnosticoOb(String rcodigoDiagnosticoOb) {
        STRcodigoDiagnosticoOb = rcodigoDiagnosticoOb;
}
public String getSTRdescricaoDiagnosticoOb() {
        return STRdescricaoDiagnosticoOb;
}
public void setSTRdescricaoDiagnosticoOb(String rdescricaoDiagnosticoOb) {
        STRdescricaoDiagnosticoOb = rdescricaoDiagnosticoOb;
}
public String getSTRnumeroDeclaracaoOb() {
        return STRnumeroDeclaracaoOb;
}
public void setSTRnumeroDeclaracaoOb(String rnumeroDeclaracaoOb) {
        STRnumeroDeclaracaoOb = rnumeroDeclaracaoOb;
}
public String[] getSTRcpfPR() {
        return STRcpfPR;
}
public void setSTRcpfPR(String[] rcpfPR) {
        STRcpfPR = rcpfPR;
}
public String[] getSTRcodigoPrestadorNaOperadoraPR() {
        return STRcodigoPrestadorNaOperadoraPR;
}
public void setSTRcodigoPrestadorNaOperadoraPR(
                String[] rcodigoPrestadorNaOperadoraPR) {
        STRcodigoPrestadorNaOperadoraPR = rcodigoPrestadorNaOperadoraPR;
}
public String getSTRsiglaConselhoPRCP() {
        return STRsiglaConselhoPRCP;
}
public void setSTRsiglaConselhoPRCP(String rsiglaConselhoPRCP) {
        STRsiglaConselhoPRCP = rsiglaConselhoPRCP;
}
public String getSTRnumeroConselhoPRCP() {
        return STRnumeroConselhoPRCP;
}
public void setSTRnumeroConselhoPRCP(String rnumeroConselhoPRCP) {
        STRnumeroConselhoPRCP = rnumeroConselhoPRCP;
}
public String getSTRufConselhoPRCP() {
        return STRufConselhoPRCP;
```

```
}
public void setSTRufConselhoPRCP(String rufConselhoPRCP) {
        STRufConselhoPRCP = rufConselhoPRCP;
}
public String[] getSTRnomeExecutanteID() {
        return STRnomeExecutanteID;
}
public void setSTRnomeExecutanteID(String[] rnomeExecutanteID) {
        STRnomeExecutanteID = rnomeExecutanteID;
}
public String[] getSTRsiglaConselhoID() {
        return STRsiglaConselhoID;
}
public void setSTRsiglaConselhoID(String[] rsiglaConselhoID) {
        STRsiglaConselhoID = rsiglaConselhoID;
}
public String[] getSTRnumeroConselhoID() {
        return STRnumeroConselhoID;
}
public void setSTRnumeroConselhoID(String[] rnumeroConselhoID) {
        STRnumeroConselhoID = rnumeroConselhoID;
}
public String[] getSTRufConselhoID() {
        return STRufConselhoID;
}
public void setSTRufConselhoID(String[] rufConselhoID) {
        STRufConselhoID = rufConselhoID;
}
public String getSTRcodigoCBOSID() {
        return STRcodigoCBOSID;
}
public void setSTRcodigoCBOSID(String rcodigoCBOSID) {
        STRcodigoCBOSID = rcodigoCBOSID;
}
public String[] getSTRcpfID() {
        return STRcpfID;
}
public void setSTRcpfID(String[] rcpfID) {
        STRcpfID = rcpfID;
}
public String[] getSTRposicaoProfissionalID() {
        return STRposicaoProfissionalID;
}
public void setSTRposicaoProfissionalID(String[] rposicaoProfissionalID) {
        STRposicaoProfissionalID = rposicaoProfissionalID;
}
public String[] getSTRcodigoPR() {
        return STRcodigoPR;
}
public void setSTRcodigoPR(String[] rcodigoPR) {
        STRcodigoPR = rcodigoPR;
}
public String[] getSTRtipoTabelaPR() {
        return STRtipoTabelaPR;
}
public void setSTRtipoTabelaPR(String[] rtipoTabelaPR) {
        STRtipoTabelaPR = rtipoTabelaPR;
}
public String[] getSTRdescricaoPR() {
        return STRdescricaoPR;
```

```
}
public void setSTRdescricaoPR(String[] rdescricaoPR) {
        STRdescricaoPR = rdescricaoPR;
}
public String[] getSTRdataPR() {
        return STRdataPR;
}
public void setSTRdataPR(String[] rdataPR) {
        STRdataPR = rdataPR;
}
public String[] getSTRhoraInicioPR() {
        return STRhoraInicioPR;
}
public void setSTRhoraInicioPR(String[] rhoraInicioPR) {
        STRhoraInicioPR = rhoraInicioPR;
}
public String[] getSTRhoraFimPR() {
        return STRhoraFimPR;
}
public void setSTRhoraFimPR(String[] rhoraFimPR) {
        STRhoraFimPR = rhoraFimPR;
}
public String[] getSTRquantidadeRealizadaPR() {
        return STRquantidadeRealizadaPR;
}
public void setSTRquantidadeRealizadaPR(String[] rquantidadeRealizadaPR) {
        STRquantidadeRealizadaPR = rquantidadeRealizadaPR;
}
public String[] getSTRviaAcessoPR() {
        return STRviaAcessoPR;
}
public void setSTRviaAcessoPR(String[] rviaAcessoPR) {
        STRviaAcessoPR = rviaAcessoPR;
}
public String[] getSTRtecnicaUtilizadaPR() {
        return STRtecnicaUtilizadaPR;
}
public void setSTRtecnicaUtilizadaPR(String[] rtecnicaUtilizadaPR) {
        STRtecnicaUtilizadaPR = rtecnicaUtilizadaPR;
}
public String[] getSTRreducaoAcrescimoPR() {
        return STRreducaoAcrescimoPR;
}
public void setSTRreducaoAcrescimoPR(String[] rreducaoAcrescimoPR) {
        STRreducaoAcrescimoPR = rreducaoAcrescimoPR;
}
public String[] getSTRvalorPR() {
        return STRvalorPR;
}
public void setSTRvalorPR(String[] rvalorPR) {
        STRvalorPR = rvalorPR;
}
public String[] getSTRvalorTotalPR() {
        return STRvalorTotalPR;
}
public void setSTRvalorTotalPR(String[] rvalorTotalPR) {
        STRvalorTotalPR = rvalorTotalPR;
}
public String[] getSTRcodigoOPM() {
        return STRcodigoOPM;
```

```
}
public void setSTRcodigoOPM(String[] rcodigoOPM) {
        STRcodigoOPM = rcodigoOPM;
}
public String[] getSTRtipoTabelaOPM() {
        return STRtipoTabelaOPM;
}
public void setSTRtipoTabelaOPM(String[] rtipoTabelaOPM) {
        STRtipoTabelaOPM = rtipoTabelaOPM;
}
public String[] getSTRdescricaoOPM() {
        return STRdescricaoOPM;
}
public void setSTRdescricaoOPM(String[] rdescricaoOPM) {
        STRdescricaoOPM = rdescricaoOPM;
}
public String[] getSTRquantidadeOPM() {
        return STRquantidadeOPM;
}
public void setSTRquantidadeOPM(String[] rquantidadeOPM) {
        STRquantidadeOPM = rquantidadeOPM;
}
public String[] getSTRcodigoBarraOPM() {
        return STRcodigoBarraOPM;
}
public void setSTRcodigoBarraOPM(String[] rcodigoBarraOPM) {
        STRcodigoBarraOPM = rcodigoBarraOPM;
}
public String[] getSTRvalorUnitarioOPM() {
        return STRvalorUnitarioOPM;
}
public void setSTRvalorUnitarioOPM(String[] rvalorUnitarioOPM) {
        STRvalorUnitarioOPM = rvalorUnitarioOPM;
}
public String[] getSTRvalorTotalOPM() {
        return STRvalorTotalOPM;
}
public void setSTRvalorTotalOPM(String[] rvalorTotalOPM) {
        STRvalorTotalOPM = rvalorTotalOPM;
}
public String getSTRvalorTotalOPMOPM() {
        return STRvalorTotalOPMOPM;
}
public void setSTRvalorTotalOPMOPM(String rvalorTotalOPMOPM) {
        STRvalorTotalOPMOPM = rvalorTotalOPMOPM;
}
public String getSTRcodigoOD() {
        return STRcodigoOD;
}
public void setSTRcodigoOD(String rcodigoOD) {
        STRcodigoOD = rcodigoOD;
}
public String getSTRtipoTabelaOD() {
        return STRtipoTabelaOD;
}
public void setSTRtipoTabelaOD(String rtipoTabelaOD) {
        STRtipoTabelaOD = rtipoTabelaOD;
}
public String getSTRdescricaoOD() {
        return STRdescricaoOD;
```

```
	}
	public void setSTRdescricaoOD(String rdescricaoOD) {
		STRdescricaoOD = rdescricaoOD;
	}
	public String getSTRtipoDespesaOD() {
		return STRtipoDespesaOD;
	}
	public void setSTRtipoDespesaOD(String rtipoDespesaOD) {
		STRtipoDespesaOD = rtipoDespesaOD;
	}
	public String getSTRdataRealizacaoOD() {
		return STRdataRealizacaoOD;
	}
	public void setSTRdataRealizacaoOD(String rdataRealizacaoOD) {
		STRdataRealizacaoOD = rdataRealizacaoOD;
	}
	public String getSTRhoraInicialOD() {
		return STRhoraInicialOD;
	}
	public void setSTRhoraInicialOD(String rhoraInicialOD) {
		STRhoraInicialOD = rhoraInicialOD;
	}
	public String getSTRhoraFinalOD() {
		return STRhoraFinalOD;
	}
	public void setSTRhoraFinalOD(String rhoraFinalOD) {
		STRhoraFinalOD = rhoraFinalOD;
	}
	public String getSTRreducaoAcrescimoOD() {
		return STRreducaoAcrescimoOD;
	}
	public void setSTRreducaoAcrescimoOD(String rreducaoAcrescimoOD) {
		STRreducaoAcrescimoOD = rreducaoAcrescimoOD;
	}
	public String getSTRquantidadeOD() {
		return STRquantidadeOD;
	}
	public void setSTRquantidadeOD(String rquantidadeOD) {
		STRquantidadeOD = rquantidadeOD;
	}
	public String getSTRvalorUnitarioOD() {
		return STRvalorUnitarioOD;
	}
	public void setSTRvalorUnitarioOD(String rvalorUnitarioOD) {
		STRvalorUnitarioOD = rvalorUnitarioOD;
	}
	public String getSTRvalorTotalOD() {
		return STRvalorTotalOD;
	}
	public void setSTRvalorTotalOD(String rvalorTotalOD) {
		STRvalorTotalOD = rvalorTotalOD;
	}
	public String getSTRtotalGeralOutrasDespesasOD() {
		return STRtotalGeralOutrasDespesasOD;
	}
	public void setSTRtotalGeralOutrasDespesasOD(String
rtotalGeralOutrasDespesasOD) {
		STRtotalGeralOutrasDespesasOD = rtotalGeralOutrasDespesasOD;
	}
	public String getSTRtipoFaturamento() {
```

```
            return STRtipoFaturamento;
        }
        public void setSTRtipoFaturamento(String rtipoFaturamento) {
            STRtipoFaturamento = rtipoFaturamento;
        }
        public String getSTRservicosExecutados() {
            return STRservicosExecutados;
        }
        public void setSTRservicosExecutados(String rservicosExecutados) {
            STRservicosExecutados = rservicosExecutados;
        }
        public String getSTRdiarias() {
            return STRdiarias;
        }
        public void setSTRdiarias(String rdiarias) {
            STRdiarias = rdiarias;
        }
        public String getSTRtaxas() {
            return STRtaxas;
        }
        public void setSTRtaxas(String rtaxas) {
            STRtaxas = rtaxas;
        }
        public String getSTRmateriais() {
            return STRmateriais;
        }
        public void setSTRmateriais(String rmateriais) {
            STRmateriais = rmateriais;
        }
        public String getSTRmedicamentos() {
            return STRmedicamentos;
        }
        public void setSTRmedicamentos(String rmedicamentos) {
            STRmedicamentos = rmedicamentos;
        }
        public String getSTRgases() {
            return STRgases;
        }
        public void setSTRgases(String rgases) {
            STRgases = rgases;
        }
        public String getSTRtotalGeral() {
            return STRtotalGeral;
        }
        public void setSTRtotalGeral(String rtotalGeral) {
            STRtotalGeral = rtotalGeral;
        }
        public String getSTRobservacao() {
            return STRobservacao;
        }
        public void setSTRobservacao(String robservacao) {
            STRobservacao = robservacao;
        }

}
```

Nome do arquivo: CIGuiaSolilnternacao.java

```
package com.classes;
```

```
import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import com.util.guia.Util;

public class ClGuiaSolilnternacao {
        private String STRregistroANS;
        private String STRdataEmissaoGuia;
        private String STRnumeroGuiaPrestador;
        private String STRnumeroGuiaOperadora;
        private String STRnumeroCarteira;
        private String STRnomeBeneficiario;
        private String STRnomePlano;
        private String STRvalidadeCarteira;
        private String STRnumeroCNS;
        private String STRidentificadorBeneficiario;
        private String STRCNPJC;
        private String STRcpfC;
        private String STRcodigoPrestadorNaOperadoraC;
        private String STRsiglaConselhoC;
        private String STRnumeroConselhoC;
        private String STRufConselhoC;
        private String STRnomeContratadoC;
        private String STRntipoLogradouroC;
        private String STRlogradouroC;
        private String STRnumeroC;
        private String STRcomplementoC;
        private String STRcodigolBGEMunicipioC;
        private String STRmunicipioC;
        private String STRcodigoUFC;
        private String STRcepC;
        private String STRnumeroCNESC;
        private String STRnomeProfissionalP;
        private String STRsiglaConselhoP;
        private String STRnumeroConselhoP;
        private String STRufConselhoP;
        private String STRcbosP;
        private String STRCNPJPS;
        private String STRcpfPS;
        private String STRcodigoPrestadorNaOperadoraPS;
        private String STRsiglaConselhoPS;
        private String STRnumeroConselhoPS;
        private String STRufConselhoPS;
        private String STRnomePrestadorPS;
        private String STRcaraterInternacao;
        private String STRtipoInternacao;
        private String STRindicacaoClinica;
        private String STRregimeInternacao;
        private String STRnomeTabela;
        private String STRcodigoDiagnostico;
        private String STRdescricaoDiagnostico;
        private String STRtipoDoenca;
        private String STRvalor;
        private String STRunidadeTempo;
        private String STRindicadorAcidente;
        private String STRnomeTabelaSEC;
        private String[] STRcodigoDiagnosticoSEC;
        private String STRdescricaoDiagnosticoSEC;
```

```
        private String[] STRquantidadeSolicitadaPES;
        private String[] STRcodigoPES;
        private String[] STRtipoTabelaPES;
        private String[] STRdescricaoPES;
        private String[] STRquantidadeSolicitadaOPM;
        private String[] STRfabricanteOPM;
        private String[] STRvalorOPM;
        private String[] STRcodigoOPM;
        private String[] STRtipoTabelaOPM;
        private String[] STRdescricaoOPM;
        private String STRdiasSolicitados;
        private String STRdataProvavelAdmisHosp;
        private String STRobservacao;

        public static ClGuiaSoliInternacao getGuiaDoRequest(HttpServletRequest request) {
                ClGuiaSoliInternacao guia = new ClGuiaSoliInternacao();

                guia.STRregistroANS    =
request.getParameter("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.registroAN
S");
                guia.STRdataEmissaoGuia      =
request.getParameter("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.dataEmissa
oGuia");
                guia.STRnumeroGuiaPrestador =
request.getParameter("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGui
aPrestador");
                guia.STRnumeroGuiaOperadora          =
request.getParameter("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGui
aOperadora");

                guia.STRnumeroCarteira          =
request.getParameter("guiaSolicInternacao.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario      =
request.getParameter("guiaSolicInternacao.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano    =
request.getParameter("guiaSolicInternacao.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira          =
request.getParameter("guiaSolicInternacao.dadosBeneficiario.validadeCarteira");
                guia.STRnumeroCNS    =
request.getParameter("guiaSolicInternacao.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario          =
request.getParameter("guiaSolicInternacao.dadosBeneficiario.identificadorBeneficiario");

                guia.STRCNPJC          =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.CNPJ");
                guia.STRcpfC    =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.cpf");
                guia.STRcodigoPrestadorNaOperadoraC          =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.codigoPr
estadorNaOperadora");
                guia.STRsiglaConselhoC          =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.conselho
Profissional.siglaConselho");
                guia.STRnumeroConselhoC      =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.conselho
Profissional.numeroConselho");
                guia.STRufConselhoC    =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.conselho
Profissional.ufConselho");
```

```
                guia.STRnomeContratadoC      =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.nomeContratado");
                guia.STRntipoLogradouroC      =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.ti
poLogradouro");
                guia.STRlogradouroC   =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.l
ogradouro");
                guia.STRnumeroC       =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.n
umero");
                guia.STRcomplementoC          =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.c
omplemento");
                guia.STRcodigoIBGEMunicipioC          =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.c
odigoIBGEMunicipio");
                guia.STRmunicipioC    =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.
municipio");
                guia.STRcodigoUFC     =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.c
odigoUF");
                guia.STRcepC  =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.c
ep");
                guia.STRnumeroCNESC           =
request.getParameter("guiaSolicInternacao.dadosSolicitante.contratado.numeroCNES");

                guia.STRnomeProfissionalP     =
request.getParameter("guiaSolicInternacao.dadosSolicitante.profissional.nomeProfissional");
                guia.STRsiglaConselhoP        =
request.getParameter("guiaSolicInternacao.dadosSolicitante.profissional.conselhoProfissional.
siglaConselho");
                guia.STRnumeroConselhoP       =
request.getParameter("guiaSolicInternacao.dadosSolicitante.profissional.conselhoProfissional.
numeroConselho");
                guia.STRufConselhoP   =
request.getParameter("guiaSolicInternacao.dadosSolicitante.profissional.conselhoProfissional.
ufConselho");
                guia.STRcbosP=
request.getParameter("guiaSolicInternacao.dadosSolicitante.profissional.cbos");

                guia.STRCNPJPS        =
request.getParameter("guiaSolicInternacao.prestadorSolicitado.CNPJ");
                guia.STRcpfPS =
request.getParameter("guiaSolicInternacao.prestadorSolicitado.cpf");
                guia.STRcodigoPrestadorNaOperadoraPS          =
request.getParameter("guiaSolicInternacao.prestadorSolicitado.codigoPrestadorNaOperadora"
);
                guia.STRsiglaConselhoPS       =
request.getParameter("guiaSolicInternacao.prestadorSolicitado.conselhoProfissional.siglaCons
elho");
                guia.STRnumeroConselhoPS      =
request.getParameter("guiaSolicInternacao.prestadorSolicitado.conselhoProfissional.numeroC
onselho");
                guia.STRufConselhoPS=
request.getParameter("guiaSolicInternacao.prestadorSolicitado.conselhoProfissional.ufConsel
ho");
```

```
                guia.STRnomePrestadorPS      =
request.getParameter("guiaSolicInternacao.prestadorSolicitado.nomePrestador");

                guia.STRcaraterInternacao      =
request.getParameter("guiaSolicInternacao.caraterInternacao");
                guia.STRtipoInternacao=
request.getParameter("guiaSolicInternacao.tipoInternacao");
                guia.STRindicacaoClinica      =
request.getParameter("guiaSolicInternacao.indicacaoClinica");
                guia.STRregimeInternacao      =
request.getParameter("guiaSolicInternacao.regimeInternacao");

                guia.STRnomeTabela  =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.CID.nomeTab
ela");
                guia.STRcodigoDiagnostico      =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.CID.codigoDia
gnostico");
                guia.STRdescricaoDiagnostico =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.CID.descricao
Diagnostico");
                guia.STRtipoDoenca    =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.tipoDoenca");
                guia.STRvalor  =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.tempoReferido
EvolucaoDoenca.valor");
                guia.STRunidadeTempo      =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.tempoReferido
EvolucaoDoenca.unidadeTempo");
                guia.STRindicadorAcidente      =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.indicadorAcide
nte");
                guia.STRnomeTabelaSEC      =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.diagnosticosSecundarios.CI
D.nomeTabela");
                guia.STRcodigoDiagnosticoSEC      =
request.getParameterValues("guiaSolicInternacao.hipotesesDiagnosticas.diagnosticosSecund
arios.CID.codigoDiagnostico");
                guia.STRdescricaoDiagnosticoSEC      =
request.getParameter("guiaSolicInternacao.hipotesesDiagnosticas.diagnosticosSecundarios.CI
D.descricaoDiagnostico");

                guia.STRquantidadeSolicitadaPES      =
request.getParameterValues("guiaSolicInternacao.procedimentosExamesSolicitados.procedim
entoSolicitado.quantidadeSolicitada");
                guia.STRcodigoPES    =
request.getParameterValues("guiaSolicInternacao.procedimentosExamesSolicitados.procedim
entoSolicitado.procedimento.codigo");
                guia.STRtipoTabelaPES      =
request.getParameterValues("guiaSolicInternacao.procedimentosExamesSolicitados.procedim
entoSolicitado.procedimento.tipoTabela");
                guia.STRdescricaoPES =
request.getParameterValues("guiaSolicInternacao.procedimentosExamesSolicitados.procedim
entoSolicitado.procedimento.descricao");

                guia.STRquantidadeSolicitadaOPM      =
request.getParameterValues("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.quantidad
eSolicitada");
```

```
                guia.STRfabricanteOPM          =
request.getParameterValues("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.fabricante"
);
                guia.STRvalorOPM      =
request.getParameterValues("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.valor");
                guia.STRcodigoOPM     =
request.getParameterValues("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.OPM.codi
go");
                guia.STRtipoTabelaOPM         =
request.getParameterValues("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.OPM.tipoT
abela");
                guia.STRdescricaoOPM          =
request.getParameterValues("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.OPM.desc
ricao");
                guia.STRdiasSolicitados       =
request.getParameter("guiaSolicInternacao.diasSolicitados");
                guia.STRdataProvavelAdmisHosp       =
request.getParameter("guiaSolicInternacao.dataProvavelAdmisHosp");
                guia.STRobservacao    =
request.getParameter("guiaSolicInternacao.observacao");

                return guia;
        }

        public static ClGuiaSoliInternacao getGuiaDoMapa(Map<String, String[]> mapa) {
                ClGuiaSoliInternacao guia = new ClGuiaSoliInternacao();

                guia.STRregistroANS    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.registroANS");
                guia.STRdataEmissaoGuia       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.dataEmissaoGuia");
                guia.STRnumeroGuiaPrestador = Util.getPrimeiro(mapa,
"guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGuiaPrestador");
                guia.STRnumeroGuiaOperadora          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGuiaOperadora");

                guia.STRnumeroCarteira          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira        = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosBeneficiario.validadeCarteira");
                guia.STRnumeroCNS    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosBeneficiario.identificadorBeneficiario");

                guia.STRCNPJC          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.identificacao.CNPJ");
                guia.STRcpfC    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.identificacao.cpf");
                guia.STRcodigoPrestadorNaOperadoraC          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.identificacao.codigoPrestadorNaOperadora")
;
                guia.STRsiglaConselhoC          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.identificacao.conselhoProfissional.siglaCons
elho");
```

```
                guia.STRnumeroConselhoC     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.identificacao.conselhoProfissional.numeroCo
nselho");
                guia.STRufConselhoC  = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.identificacao.conselhoProfissional.ufConselh
o");
                guia.STRnomeContratadoC     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.nomeContratado");
                guia.STRntipoLogradouroC     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.tipoLogradouro");
                guia.STRlogradouroC  = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.logradouro");
                guia.STRnumeroC      = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.numero");
                guia.STRcomplementoC         = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.complemento");
                guia.STRcodigoIBGEMunicipioC         = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.codigoIBGEMunicipio");
                guia.STRmunicipioC   = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.municipio");
                guia.STRcodigoUFC    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.codigoUF");
                guia.STRcepC  = Util.getPrimeiro(mapa, .
"guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.cep");
                guia.STRnumeroCNESC          = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.contratado.numeroCNES");

                guia.STRnomeProfissionalP    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.profissional.nomeProfissional");
                guia.STRsiglaConselhoP       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.profissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoP      = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.profissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoP  = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.profissional.conselhoProfissional.ufConselho");
                guia.STRcbosP = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dadosSolicitante.profissional.cbos");

                guia.STRCNPJPS       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.CNPJ");
                guia.STRcpfPS = Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.cpf");
                guia.STRcodigoPrestadorNaOperadoraPS      = Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.codigoPrestadorNaOperadora");
                guia.STRsiglaConselhoPS      = Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoPS     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.conselhoProfissional.numeroConselho");
                guia.STRufConselhoPS= Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.conselhoProfissional.ufConselho");
                guia.STRnomePrestadorPS      = Util.getPrimeiro(mapa,
"guiaSolicInternacao.prestadorSolicitado.nomePrestador");

                guia.STRcaraterInternacao    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.caraterInternacao");
                guia.STRtipoInternacao= Util.getPrimeiro(mapa,
"guiaSolicInternacao.tipoInternacao");
                guia.STRindicacaoClinica     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.indicacaoClinica");
```

```
                guia.STRregimeInternacao        = Util.getPrimeiro(mapa,
"guiaSolicInternacao.regimeInternacao");

                guia.STRnomeTabela   = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.CID.nomeTabela");
                guia.STRcodigoDiagnostico       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.CID.codigoDiagnostico");
                guia.STRdescricaoDiagnostico  = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.CID.descricaoDiagnostico");
                guia.STRtipoDoenca     = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.tipoDoenca");
                guia.STRvalor   = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.tempoReferidoEvolucaoDoenca.valor
");
                guia.STRunidadeTempo            = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.tempoReferidoEvolucaoDoenca.unid
adeTempo");
                guia.STRindicadorAcidente       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.CIDPrincipal.indicadorAcidente");
                guia.STRnomeTabelaSEC       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.diagnosticosSecundarios.CID.nomeTabela");
                guia.STRcodigoDiagnosticoSEC            =
mapa.get("guiaSolicInternacao.hipotesesDiagnosticas.diagnosticosSecundarios.CID.codigoDia
gnostico");
                guia.STRdescricaoDiagnosticoSEC      = Util.getPrimeiro(mapa,
"guiaSolicInternacao.hipotesesDiagnosticas.diagnosticosSecundarios.CID.descricaoDiagnostic
o");

                guia.STRquantidadeSolicitadaPES       =
mapa.get("guiaSolicInternacao.procedimentosExamesSolicitados.procedimentoSolicitado.quan
tidadeSolicitada");
                guia.STRcodigoPES     =
mapa.get("guiaSolicInternacao.procedimentosExamesSolicitados.procedimentoSolicitado.proc
edimento.codigo");
                guia.STRtipoTabelaPES           =
mapa.get("guiaSolicInternacao.procedimentosExamesSolicitados.procedimentoSolicitado.proc
edimento.tipoTabela");
                guia.STRdescricaoPES =
mapa.get("guiaSolicInternacao.procedimentosExamesSolicitados.procedimentoSolicitado.proc
edimento.descricao");

                guia.STRquantidadeSolicitadaOPM       =
mapa.get("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.quantidadeSolicitada");
                guia.STRfabricanteOPM           =
mapa.get("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.fabricante");
                guia.STRvalorOPM      =
mapa.get("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.valor");
                guia.STRcodigoOPM     =
mapa.get("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.OPM.codigo");
                guia.STRtipoTabelaOPM           =
mapa.get("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.OPM.tipoTabela");
                guia.STRdescricaoOPM            =
mapa.get("guiaSolicInternacao.OPMsSolicitadas.OPMSolicitada.OPM.descricao");
                guia.STRdiasSolicitados         = Util.getPrimeiro(mapa,
"guiaSolicInternacao.diasSolicitados");
                guia.STRdataProvavelAdmisHosp       = Util.getPrimeiro(mapa,
"guiaSolicInternacao.dataProvavelAdmisHosp");
                guia.STRobservacao    = Util.getPrimeiro(mapa,
"guiaSolicInternacao.observacao");
```

```
        return guia;
}


public String getSTRregistroANS() {
        return STRregistroANS;
}
public void setSTRregistroANS(String rregistroANS) {
        STRregistroANS = rregistroANS;
}
public String getSTRdataEmissaoGuia() {
        return STRdataEmissaoGuia;
}
public void setSTRdataEmissaoGuia(String rdataEmissaoGuia) {
        STRdataEmissaoGuia = rdataEmissaoGuia;
}
public String getSTRnumeroGuiaPrestador() {
        return STRnumeroGuiaPrestador;
}
public void setSTRnumeroGuiaPrestador(String rnumeroGuiaPrestador) {
        STRnumeroGuiaPrestador = rnumeroGuiaPrestador;
}
public String getSTRnumeroGuiaOperadora() {
        return STRnumeroGuiaOperadora;
}
public void setSTRnumeroGuiaOperadora(String rnumeroGuiaOperadora) {
        STRnumeroGuiaOperadora = rnumeroGuiaOperadora;
}
public String getSTRnumeroCarteira() {
        return STRnumeroCarteira;
}
public void setSTRnumeroCarteira(String rnumeroCarteira) {
        STRnumeroCarteira = rnumeroCarteira;
}
public String getSTRnomeBeneficiario() {
        return STRnomeBeneficiario;
}
public void setSTRnomeBeneficiario(String rnomeBeneficiario) {
        STRnomeBeneficiario = rnomeBeneficiario;
}
public String getSTRnomePlano() {
        return STRnomePlano;
}
public void setSTRnomePlano(String rnomePlano) {
        STRnomePlano = rnomePlano;
}
public String getSTRvalidadeCarteira() {
        return STRvalidadeCarteira;
}
public void setSTRvalidadeCarteira(String rvalidadeCarteira) {
        STRvalidadeCarteira = rvalidadeCarteira;
}
public String getSTRnumeroCNS() {
        return STRnumeroCNS;
}
public void setSTRnumeroCNS(String rnumeroCNS) {
        STRnumeroCNS = rnumeroCNS;
}
public String getSTRidentificadorBeneficiario() {
        return STRidentificadorBeneficiario;
```

```
}
public void setSTRidentificadorBeneficiario(String ridentificadorBeneficiario) {
        STRidentificadorBeneficiario = ridentificadorBeneficiario;
}
public String getSTRCNPJC() {
        return STRCNPJC;
}
public void setSTRCNPJC(String strcnpjc) {
        STRCNPJC = strcnpjc;
}
public String getSTRcpfC() {
        return STRcpfC;
}
public void setSTRcpfC(String rcpfC) {
        STRcpfC = rcpfC;
}
public String getSTRcodigoPrestadorNaOperadoraC() {
        return STRcodigoPrestadorNaOperadoraC;
}
public void setSTRcodigoPrestadorNaOperadoraC(
                String rcodigoPrestadorNaOperadoraC) {
        STRcodigoPrestadorNaOperadoraC = rcodigoPrestadorNaOperadoraC;
}
public String getSTRsiglaConselhoC() {
        return STRsiglaConselhoC;
}
public void setSTRsiglaConselhoC(String rsiglaConselhoC) {
        STRsiglaConselhoC = rsiglaConselhoC;
}
public String getSTRnumeroConselhoC() {
        return STRnumeroConselhoC;
}
public void setSTRnumeroConselhoC(String rnumeroConselhoC) {
        STRnumeroConselhoC = rnumeroConselhoC;
}
public String getSTRufConselhoC() {
        return STRufConselhoC;
}
public void setSTRufConselhoC(String rufConselhoC) {
        STRufConselhoC = rufConselhoC;
}
public String getSTRnomeContratadoC() {
        return STRnomeContratadoC;
}
public void setSTRnomeContratadoC(String rnomeContratadoC) {
        STRnomeContratadoC = rnomeContratadoC;
}
public String getSTRntipoLogradouroC() {
        return STRntipoLogradouroC;
}
public void setSTRntipoLogradouroC(String rntipoLogradouroC) {
        STRntipoLogradouroC = rntipoLogradouroC;
}
public String getSTRlogradouroC() {
        return STRlogradouroC;
}
public void setSTRlogradouroC(String rlogradouroC) {
        STRlogradouroC = rlogradouroC;
}
public String getSTRnumeroC() {
```

```
        return STRnumeroC;
}
public void setSTRnumeroC(String rnumeroC) {
        STRnumeroC = rnumeroC;
}
public String getSTRcomplementoC() {
        return STRcomplementoC;
}
public void setSTRcomplementoC(String rcomplementoC) {
        STRcomplementoC = rcomplementoC;
}
public String getSTRcodigoIBGEMunicipioC() {
        return STRcodigoIBGEMunicipioC;
}
public void setSTRcodigoIBGEMunicipioC(String rcodigoIBGEMunicipioC) {
        STRcodigoIBGEMunicipioC = rcodigoIBGEMunicipioC;
}
public String getSTRmunicipioC() {
        return STRmunicipioC;
}
public void setSTRmunicipioC(String rmunicipioC) {
        STRmunicipioC = rmunicipioC;
}
public String getSTRcodigoUFC() {
        return STRcodigoUFC;
}
public void setSTRcodigoUFC(String rcodigoUFC) {
        STRcodigoUFC = rcodigoUFC;
}
public String getSTRcepC() {
        return STRcepC;
}
public void setSTRcepC(String rcepC) {
        STRcepC = rcepC;
}
public String getSTRnumeroCNESC() {
        return STRnumeroCNESC;
}
public void setSTRnumeroCNESC(String rnumeroCNESC) {
        STRnumeroCNESC = rnumeroCNESC;
}
public String getSTRnomeProfissionalP() {
        return STRnomeProfissionalP;
}
public void setSTRnomeProfissionalP(String rnomeProfissionalP) {
        STRnomeProfissionalP = rnomeProfissionalP;
}
public String getSTRsiglaConselhoP() {
        return STRsiglaConselhoP;
}
public void setSTRsiglaConselhoP(String rsiglaConselhoP) {
        STRsiglaConselhoP = rsiglaConselhoP;
}
public String getSTRnumeroConselhoP() {
        return STRnumeroConselhoP;
}
public void setSTRnumeroConselhoP(String rnumeroConselhoP) {
        STRnumeroConselhoP = rnumeroConselhoP;
}
public String getSTRufConselhoP() {
```

```
        return STRufConselhoP;
}
public void setSTRufConselhoP(String rufConselhoP) {
        STRufConselhoP = rufConselhoP;
}
public String getSTRcbosP() {
        return STRcbosP;
}
public void setSTRcbosP(String rcbosP) {
        STRcbosP = rcbosP;
}
public String getSTRCNPJPS() {
        return STRCNPJPS;
}
public void setSTRCNPJPS(String strcnpjps) {
        STRCNPJPS = strcnpjps;
}
public String getSTRcpfPS() {
        return STRcpfPS;
}
public void setSTRcpfPS(String rcpfPS) {
        STRcpfPS = rcpfPS;
}
public String getSTRcodigoPrestadorNaOperadoraPS() {
        return STRcodigoPrestadorNaOperadoraPS;
}
public void setSTRcodigoPrestadorNaOperadoraPS(
                String rcodigoPrestadorNaOperadoraPS) {
        STRcodigoPrestadorNaOperadoraPS = rcodigoPrestadorNaOperadoraPS;
}
public String getSTRsiglaConselhoPS() {
        return STRsiglaConselhoPS;
}
public void setSTRsiglaConselhoPS(String rsiglaConselhoPS) {
        STRsiglaConselhoPS = rsiglaConselhoPS;
}
public String getSTRnumeroConselhoPS() {
        return STRnumeroConselhoPS;
}
public void setSTRnumeroConselhoPS(String rnumeroConselhoPS) {
        STRnumeroConselhoPS = rnumeroConselhoPS;
}
public String getSTRufConselhoPS() {
        return STRufConselhoPS;
}
public void setSTRufConselhoPS(String rufConselhoPS) {
        STRufConselhoPS = rufConselhoPS;
}
public String getSTRnomePrestadorPS() {
        return STRnomePrestadorPS;
}
public void setSTRnomePrestadorPS(String rnomePrestadorPS) {
        STRnomePrestadorPS = rnomePrestadorPS;
}
public String getSTRcaraterInternacao() {
        return STRcaraterInternacao;
}
public void setSTRcaraterInternacao(String rcaraterInternacao) {
        STRcaraterInternacao = rcaraterInternacao;
}
```

```
public String getSTRtipoInternacao() {
        return STRtipoInternacao;
}
public void setSTRtipoInternacao(String rtipoInternacao) {
        STRtipoInternacao = rtipoInternacao;
}
public String getSTRindicacaoClinica() {
        return STRindicacaoClinica;
}
public void setSTRindicacaoClinica(String rindicacaoClinica) {
        STRindicacaoClinica = rindicacaoClinica;
}
public String getSTRregimeInternacao() {
        return STRregimeInternacao;
}
public void setSTRregimeInternacao(String rregimeInternacao) {
        STRregimeInternacao = rregimeInternacao;
}
public String getSTRnomeTabela() {
        return STRnomeTabela;
}
public void setSTRnomeTabela(String rnomeTabela) {
        STRnomeTabela = rnomeTabela;
}
public String getSTRcodigoDiagnostico() {
        return STRcodigoDiagnostico;
}
public void setSTRcodigoDiagnostico(String rcodigoDiagnostico) {
        STRcodigoDiagnostico = rcodigoDiagnostico;
}
public String getSTRdescricaoDiagnostico() {
        return STRdescricaoDiagnostico;
}
public void setSTRdescricaoDiagnostico(String rdescricaoDiagnostico) {
        STRdescricaoDiagnostico = rdescricaoDiagnostico;
}
public String getSTRtipoDoenca() {
        return STRtipoDoenca;
}
public void setSTRtipoDoenca(String rtipoDoenca) {
        STRtipoDoenca = rtipoDoenca;
}
public String getSTRvalor() {
        return STRvalor;
}
public void setSTRvalor(String rvalor) {
        STRvalor = rvalor;
}
public String getSTRunidadeTempo() {
        return STRunidadeTempo;
}
public void setSTRunidadeTempo(String runidadeTempo) {
        STRunidadeTempo = runidadeTempo;
}
public String getSTRindicadorAcidente() {
        return STRindicadorAcidente;
}
public void setSTRindicadorAcidente(String rindicadorAcidente) {
        STRindicadorAcidente = rindicadorAcidente;
}
```

```
public String getSTRnomeTabelaSEC() {
        return STRnomeTabelaSEC;
}
public void setSTRnomeTabelaSEC(String rnomeTabelaSEC) {
        STRnomeTabelaSEC = rnomeTabelaSEC;
}
public String[] getSTRcodigoDiagnosticoSEC() {
        return STRcodigoDiagnosticoSEC;
}
public void setSTRcodigoDiagnosticoSEC(String[] rcodigoDiagnosticoSEC) {
        STRcodigoDiagnosticoSEC = rcodigoDiagnosticoSEC;
}
public String getSTRdescricaoDiagnosticoSEC() {
        return STRdescricaoDiagnosticoSEC;
}
public void setSTRdescricaoDiagnosticoSEC(String rdescricaoDiagnosticoSEC) {
        STRdescricaoDiagnosticoSEC = rdescricaoDiagnosticoSEC;
}
public String[] getSTRquantidadeSolicitadaPES() {
        return STRquantidadeSolicitadaPES;
}
public void setSTRquantidadeSolicitadaPES(String[] rquantidadeSolicitadaPES) {
        STRquantidadeSolicitadaPES = rquantidadeSolicitadaPES;
}
public String[] getSTRcodigoPES() {
        return STRcodigoPES;
}
public void setSTRcodigoPES(String[] rcodigoPES) {
        STRcodigoPES = rcodigoPES;
}
public String[] getSTRtipoTabelaPES() {
        return STRtipoTabelaPES;
}
public void setSTRtipoTabelaPES(String[] rtipoTabelaPES) {
        STRtipoTabelaPES = rtipoTabelaPES;
}
public String[] getSTRdescricaoPES() {
        return STRdescricaoPES;
}
public void setSTRdescricaoPES(String[] rdescricaoPES) {
        STRdescricaoPES = rdescricaoPES;
}
public String[] getSTRquantidadeSolicitadaOPM() {
        return STRquantidadeSolicitadaOPM;
}
public void setSTRquantidadeSolicitadaOPM(String[] rquantidadeSolicitadaOPM) {
        STRquantidadeSolicitadaOPM = rquantidadeSolicitadaOPM;
}
public String[] getSTRfabricanteOPM() {
        return STRfabricanteOPM;
}
public void setSTRfabricanteOPM(String[] rfabricanteOPM) {
        STRfabricanteOPM = rfabricanteOPM;
}
public String[] getSTRvalorOPM() {
        return STRvalorOPM;
}
public void setSTRvalorOPM(String[] rvalorOPM) {
        STRvalorOPM = rvalorOPM;
}
```

```
	public String[] getSTRcodigoOPM() {
		return STRcodigoOPM;
	}
	public void setSTRcodigoOPM(String[] rcodigoOPM) {
		STRcodigoOPM = rcodigoOPM;
	}
	public String[] getSTRtipoTabelaOPM() {
		return STRtipoTabelaOPM;
	}
	public void setSTRtipoTabelaOPM(String[] rtipoTabelaOPM) {
		STRtipoTabelaOPM = rtipoTabelaOPM;
	}
	public String[] getSTRdescricaoOPM() {
		return STRdescricaoOPM;
	}
	public void setSTRdescricaoOPM(String[] rdescricaoOPM) {
		STRdescricaoOPM = rdescricaoOPM;
	}
	public String getSTRdiasSolicitados() {
		return STRdiasSolicitados;
	}
	public void setSTRdiasSolicitados(String rdiasSolicitados) {
		STRdiasSolicitados = rdiasSolicitados;
	}
	public String getSTRdataProvavelAdmisHosp() {
		return STRdataProvavelAdmisHosp;
	}
	public void setSTRdataProvavelAdmisHosp(String rdataProvavelAdmisHosp) {
		STRdataProvavelAdmisHosp = rdataProvavelAdmisHosp;
	}
	public String getSTRobservacao() {
		return STRobservacao;
	}
	public void setSTRobservacao(String robservacao) {
		STRobservacao = robservacao;
	}
}
```

Nome do arquivo: ClGuiaSPSADT.java

```
package com.classes;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import com.util.guia.Util;

public class ClGuiaSPSADT {
	private String STRregistroANS;
	private String STRdataEmissaoGuia;
	private String STRnumeroGuiaPrestador;
	private String STRnumeroGuiaOperadora;
	private String STRnumeroGuiaPrincipal;
	private String STRdataAutorizacao;
	private String STRsenhaAutorizacao;
	private String STRvalidadeSenha;
	private String STRnumeroCarteira;
	private String STRnomeBeneficiario;
```

```
private String STRnomePlano;
private String STRvalidadeCarteira;
private String STRnumeroCNS;
private String STRidentificadorBeneficiario;
private String STRCNPJC;
private String STRcpfC;
private String STRcodigoPrestadorNaOperadoraC;
private String STRsiglaConselhoC;
private String STRnumeroConselhoC;
private String STRufConselhoC;
private String STRnomeContratadoC;
private String STRntipoLogradouroC;
private String STRlogradouroC;
private String STRnumeroC;
private String STRcomplementoC;
private String STRcodigoIBGEMunicipioC;
private String STRmunicipioC;
private String STRcodigoUFC;
private String STRcepC;
private String STRnumeroCNESC;
private String STRnomeProfissionalP;
private String STRsiglaConselhoP;
private String STRnumeroConselhoP;
private String STRufConselhoP;
private String STRcbosP;
private String STRCNPJPE;
private String STRCPFPE;
private String STRcodigoPrestadorNaOperadoraPE;
private String STRnomeContratadoPE;
private String STRtipoLogradouroPE;
private String STRlogradouroPE;
private String STRnumeroPE;
private String STRcomplementoPE;
private String STRcodigoIBGEMunicipioPE;
private String STRmunicipioPE;
private String STRcodigoUFPE;
private String STRcepPE;
private String STRnumeroCNESPE;
private String STRnomeExecutantePEC;
private String STRsiglaConselhoPEC;
private String STRnumeroConselhoPEC;
private String STRufConselhoPEC;
private String STRcodigoCBOSPEC;
private String STRcpfPEC;
private String STRcodigoPrestadorNaOperadoraPEC;
private String STRindicacaoClinica;
private String STRcaraterAtendimento;
private String STRdataHoraAtendimento;
private String STRnomeTabela;
private String STRcodigoDiagnostico;
private String STRdescricaoDiagnostico;
private String STRtipoDoenca;
private String STRvalor;
private String STRunidadeTempo;
private String STRindicadorAcidente;
private String STRtipoSaida;
private String STRtipoAtendimento;
private String STRcpfPR;
private String STRcodigoPrestadorNaOperadoraPR;
private String STRsiglaConselhoPRCP;
```

```
	private String STRnumeroConselhoPRCP;
	private String STRufConselhoPRCP;
	private String STRnomeExecutanteID;
	private String STRsiglaConselhoID;
	private String STRnumeroConselhoID;
	private String STRufConselhoID;
	private String STRcodigoCBOSID;
	private String STRcpfID;
	private String STRposicaoProfissionalID;
	private String[] STRcodigoPR;
	private String[] STRtipoTabelaPR;
	private String[] STRdescricaoPR;
	private String[] STRdataPR;
	private String[] STRhoraInicioPR;
	private String[] STRhoraFimPR;
	private String[] STRquantidadeRealizadaPR;
	private String[] STRviaAcessoPR;
	private String[] STRtecnicaUtilizadaPR;
	private String[] STRreducaoAcrescimoPR;
	private String[] STRvalorPR;
	private String[] STRvalorTotalPR;
	private String STRcodigoOD;
	private String STRtipoTabelaOD;
	private String STRdescricaoOD;
	private String STRtipoDespesaOD;
	private String STRdataRealizacaoOD;
	private String STRhoraInicialOD;
	private String STRhoraFinalOD;
	private String STRreducaoAcrescimoOD;
	private String STRquantidadeOD;
	private String STRvalorUnitarioOD;
	private String STRvalorTotalOD;
	private String STRtotalGeralOutrasDespesasOD;
	private String[] STRcodigoOPM;
	private String[] STRtipoTabelaOPM;
	private String[] STRdescricaoOPM;
	private String[] STRquantidadeOPM;
	private String[] STRcodigoBarraOPM;
	private String[] STRvalorUnitarioOPM;
	private String[] STRvalorTotalOPM;
	private String STRvalorTotalOPMOPM;
	private String STRservicosExecutados;
	private String STRdiarias;
	private String STRtaxas;
	private String STRmateriais;
	private String STRmedicamentos;
	private String STRgases;
	private String STRtotalGeral;
	private String STRobservacao;


	public static ClGuiaSPSADT getGuiaDoRequest(HttpServletRequest request) {

		ClGuiaSPSADT guia = new ClGuiaSPSADT();

		guia.STRregistroANS	=
request.getParameter("guiaSP_SADT.identificacaoGuiaSADTSP.registroANS");
		guia.STRdataEmissaoGuia	=
request.getParameter("guiaSP_SADT.identificacaoGuiaSADTSP.dataEmissaoGuia");
```

```
guia.STRnumeroGuiaPrestador =
request.getParameter("guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaPrestador");
guia.STRnumeroGuiaOperadora =
request.getParameter("guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaOperadora");


guia.STRnumeroGuiaPrincipal =
request.getParameter("guiaSP_SADT.numeroGuiaPrincipal");


guia.STRdataAutorizacao =
request.getParameter("guiaSP_SADT.dadosAutorizacao.dataAutorizacao");
guia.STRsenhaAutorizacao =
request.getParameter("guiaSP_SADT.dadosAutorizacao.senhaAutorizacao");
guia.STRvalidadeSenha =
request.getParameter("guiaSP_SADT.dadosAutorizacao.validadeSenha");


guia.STRnumeroCarteira =
request.getParameter("guiaSP_SADT.dadosBeneficiario.numeroCarteira");
guia.STRnomeBeneficiario =
request.getParameter("guiaSP_SADT.dadosBeneficiario.nomeBeneficiario");
guia.STRnomePlano =
request.getParameter("guiaSP_SADT.dadosBeneficiario.nomePlano");
guia.STRvalidadeCarteira =
request.getParameter("guiaSP_SADT.dadosBeneficiario.validadeCarteira");
guia.STRnumeroCNS =
request.getParameter("guiaSP_SADT.dadosBeneficiario.numeroCNS");
guia.STRidentificadorBeneficiario =
request.getParameter("guiaSP_SADT.dadosBeneficiario.identificadorBeneficiario");


guia.STRCNPJC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.identificacao.CNPJ");
guia.STRcpfC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.identificacao.cpf");
guia.STRcodigoPrestadorNaOperadoraC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.identificacao.codigoPrestad
orNaOperadora");

guia.STRsiglaConselhoC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.identificacao.conselhoProfis
sional.siglaConselho");
guia.STRnumeroConselhoC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.identificacao.conselhoProfis
sional.numeroConselho");
guia.STRufConselhoC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.identificacao.conselhoProfis
sional.ufConselho");
guia.STRnomeContratadoC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.nomeContratado");

guia.STRntipoLogradouroC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.tipoLo
gradouro");
guia.STRlogradouroC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.lograd
ouro");
```

```
                guia.STRnumeroC             =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.numer
o");
                guia.STRcomplementoC        =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.compl
emento");
                guia.STRcodigoIBGEMunicipioC =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigo
IBGEMunicipio");
                guia.STRmunicipioC          =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.munici
pio");
                guia.STRcodigoUFC           =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigo
UF");
                guia.STRcepC                =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.cep");
                guia.STRnumeroCNESC         =
request.getParameter("guiaSP_SADT.dadosSolicitante.contratado.numeroCNES");

                guia.STRnomeProfissionalP =
request.getParameter("guiaSP_SADT.dadosSolicitante.profissional.nomeProfissional");

                guia.STRsiglaConselhoP  =
request.getParameter("guiaSP_SADT.dadosSolicitante.profissional.conselhoProfissional.sigla
Conselho");
                guia.STRnumeroConselhoP =
request.getParameter("guiaSP_SADT.dadosSolicitante.profissional.conselhoProfissional.nume
roConselho");
                guia.STRufConselhoP     =
request.getParameter("guiaSP_SADT.dadosSolicitante.profissional.conselhoProfissional.ufCon
selho");
                guia.STRcbosP           =
request.getParameter("guiaSP_SADT.dadosSolicitante.profissional.cbos");


                guia.STRCNPJPE                  =
request.getParameter("guiaSP_SADT.prestadorExecutante.identificacao.CNPJ");
                guia.STRCPFPE                   =
request.getParameter("guiaSP_SADT.prestadorExecutante.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadoraPE =
request.getParameter("guiaSP_SADT.prestadorExecutante.identificacao.codigoPrestadorNaO
peradora");
                guia.STRnomeContratadoPE        =
request.getParameter("guiaSP_SADT.prestadorExecutante.nomeContratado");

                guia.STRtipoLogradouroPE    =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.tipoLogradour
o");
                guia.STRlogradouroPE        =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.logradouro");
                guia.STRnumeroPE            =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.numero");
                guia.STRcomplementoPE       =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.complemento
");
                guia.STRcodigoIBGEMunicipioPE =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.codigoIBGEM
unicipio");
```

```
                guia.STRmunicipioPE          =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.municipio");
                guia.STRcodigoUFPE           =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.codigoUF");
                guia.STRcepPE                =
request.getParameter("guiaSP_SADT.prestadorExecutante.enderecoContratado.cep");
                guia.STRnumeroCNESPE         =
request.getParameter("guiaSP_SADT.prestadorExecutante.numeroCNES");

                guia.STRnomeExecutantePEC =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.nom
eExecutante");

                guia.STRsiglaConselhoPEC  =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.cons
elhoProfissional.siglaConselho");
                guia.STRnumeroConselhoPEC =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.cons
elhoProfissional.numeroConselho");
                guia.STRufConselhoPEC     =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.cons
elhoProfissional.ufConselho");
                guia.STRcodigoCBOSPEC     =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.codi
goCBOS");

                guia.STRcpfPEC                =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.codi
goProfissionalCompl.cpf");
                guia.STRcodigoPrestadorNaOperadoraPEC =
request.getParameter("guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.codi
goProfissionalCompl.codigoPrestadorNaOperadora");


                guia.STRindicacaoClinica =
request.getParameter("guiaSP_SADT.indicacaoClinica");


                guia.STRcaraterAtendimento =
request.getParameter("guiaSP_SADT.caraterAtendimento");


                guia.STRdataHoraAtendimento =
request.getParameter("guiaSP_SADT.dataHoraAtendimento");


                guia.STRnomeTabela           =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.CID.nomeTabela");
                guia.STRcodigoDiagnostico    =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.CID.codigoDiagnostico");
                guia.STRdescricaoDiagnostico =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.CID.descricaoDiagnostico");
                guia.STRtipoDoenca           =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.tipoDoenca");

                guia.STRvalor            =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.tempoReferidoEvolucaoDoenca
.valor");
```

```
                guia.STRunidadeTempo     =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.tempoReferidoEvolucaoDoenca
.unidadeTempo");
                guia.STRindicadorAcidente =
request.getParameter("guiaSP_SADT.diagnosticoAtendimento.indicadorAcidente");


                guia.STRtipoSaida = request.getParameter("guiaSP_SADT.tipoSaida");


                guia.STRtipoAtendimento =
request.getParameter("guiaSP_SADT.tipoAtendimento");


                guia.STRcpfPR                =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.codigoProfissional.cpf");
                guia.STRcodigoPrestadorNaOperadoraPR =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.codigoProfissional.codigoPrestadorNaOperadora");

                guia.STRsiglaConselhoPRCP  =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.codigoProfissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoPRCP =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.codigoProfissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoPRCP     =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.codigoProfissional.conselhoProfissional.ufConselho");

                guia.STRnomeExecutanteID =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.identificacaoProfissional.nomeExecutante");

                guia.STRsiglaConselhoID      =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.identificacaoProfissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoID     =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.identificacaoProfissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoID         =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.identificacaoProfissional.conselhoProfissional.ufConselho");
                guia.STRcodigoCBOSID         =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.identificacaoProfissional.codigoCBOS");
                guia.STRcpfID                =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.cpf");
                guia.STRposicaoProfissionalID =
request.getParameter("guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membr
oEquipe.posicaoProfissional");

            guia.STRcodigoPR             =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.procedi
mento.codigo");
                guia.STRtipoTabelaPR         =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.procedi
mento.tipoTabela");
```

```
                guia.STRdescricaoPR         =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.procedi
mento.descricao");
                guia.STRdataPR              =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.data");
                guia.STRhoraInicioPR        =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.horaIni
cio");
                guia.STRhoraFimPR           =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.horaFi
m");
                guia.STRquantidadeRealizadaPR =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.quantid
adeRealizada");
                guia.STRviaAcessoPR         =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.viaAce
sso");
                guia.STRtecnicaUtilizadaPR  =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.tecnica
Utilizada");
                guia.STRreducaoAcrescimoPR  =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.reduca
oAcrescimo");
                guia.STRvalorPR             =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.valor");
                guia.STRvalorTotalPR        =
request.getParameterValues("guiaSP_SADT.procedimentosRealizados.procedimentos.valorTo
tal");


                guia.STRcodigoOD            =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.identificadorDespesa.codigo")
;
                guia.STRtipoTabelaOD        =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.identificadorDespesa.tipoTabe
la");
                guia.STRdescricaoOD         =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.identificadorDespesa.descrica
o");
                guia.STRtipoDespesaOD       =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.tipoDespesa");
                guia.STRdataRealizacaoOD    =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.dataRealizacao");
                guia.STRhoraInicialOD       =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.horaInicial");
                guia.STRhoraFinalOD         =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.horaFinal");
                guia.STRreducaoAcrescimoOD  =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.reducaoAcrescimo");
                guia.STRquantidadeOD        =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.quantidade");
                guia.STRvalorUnitarioOD     =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.valorUnitario");
                guia.STRvalorTotalOD        =
request.getParameter("guiaSP_SADT.outrasDespesas.despesa.valorTotal");
                guia.STRtotalGeralOutrasDespesasOD =
request.getParameter("guiaSP_SADT.outrasDespesas.totalGeralOutrasDespesas");
```

```
                guia.STRcodigoOPM       =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.OPM.codig
o");
                guia.STRtipoTabelaOPM   =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.OPM.tipoT
abela");
                guia.STRdescricaoOPM    =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.OPM.desc
ricao");
                guia.STRquantidadeOPM   =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.quantidade
");
                guia.STRcodigoBarraOPM  =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.codigoBarr
a");
                guia.STRvalorUnitarioOPM =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.valorUnitar
io");
                guia.STRvalorTotalOPM   =
request.getParameterValues("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.valorTotal"
);

                guia.STRvalorTotalOPMOPM =
request.getParameter("guiaSP_SADT.OPMUtilizada.valorTotalOPM");


                guia.STRservicosExecutados =
request.getParameter("guiaSP_SADT.valorTotal.servicosExecutados");
                guia.STRdiarias         =
request.getParameter("guiaSP_SADT.valorTotal.diarias");
                guia.STRtaxas           =
request.getParameter("guiaSP_SADT.valorTotal.taxas");
                guia.STRmateriais       =
request.getParameter("guiaSP_SADT.valorTotal.materiais");
                guia.STRmedicamentos    =
request.getParameter("guiaSP_SADT.valorTotal.medicamentos");
                guia.STRgases           =
request.getParameter("guiaSP_SADT.valorTotal.gases");
                guia.STRtotalGeral      =
request.getParameter("guiaSP_SADT.valorTotal.totalGeral");

                guia.STRobservacao = request.getParameter("guiaSP_SADT.observacao");

                return guia;
        }

        public static ClGuiaSPSADT getGuiaDoMapa(Map<String, String[]> mapa) {

                ClGuiaSPSADT guia = new ClGuiaSPSADT();

                guia.STRregistroANS       = Util.getPrimeiro(mapa,
"guiaSP_SADT.identificacaoGuiaSADTSP.registroANS");
                guia.STRdataEmissaoGuia   = Util.getPrimeiro(mapa,
"guiaSP_SADT.identificacaoGuiaSADTSP.dataEmissaoGuia");
                guia.STRnumeroGuiaPrestador = Util.getPrimeiro(mapa,
"guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaPrestador");
                guia.STRnumeroGuiaOperadora = Util.getPrimeiro(mapa,
"guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaOperadora");
```

```
                guia.STRnumeroGuiaPrincipal = Util.getPrimeiro(mapa,
"guiaSP_SADT.numeroGuiaPrincipal");


                guia.STRdataAutorizacao  = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosAutorizacao.dataAutorizacao");
                guia.STRsenhaAutorizacao = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosAutorizacao.senhaAutorizacao");
                guia.STRvalidadeSenha    = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosAutorizacao.validadeSenha");


                guia.STRnumeroCarteira          = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosBeneficiario.numeroCarteira");
                guia.STRnomeBeneficiario        = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosBeneficiario.nomeBeneficiario");
                guia.STRnomePlano               = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosBeneficiario.nomePlano");
                guia.STRvalidadeCarteira        = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosBeneficiario.validadeCarteira");
                guia.STRnumeroCNS               = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosBeneficiario.numeroCNS");
                guia.STRidentificadorBeneficiario = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosBeneficiario.identificadorBeneficiario");


                guia.STRCNPJC                   = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.identificacao.CNPJ");
                guia.STRcpfC                    = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.identificacao.cpf");
                guia.STRcodigoPrestadorNaOperadoraC = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.identificacao.codigoPrestadorNaOperadora");

                guia.STRsiglaConselhoC  = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.identificacao.conselhoProfissional.siglaConselho")
;
                guia.STRnumeroConselhoC = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.identificacao.conselhoProfissional.numeroConselh
o");
                guia.STRufConselhoC     = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.identificacao.conselhoProfissional.ufConselho");
                guia.STRnomeContratadoC = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.nomeContratado");

                guia.STRntipoLogradouroC     = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.tipoLogradouro");
                guia.STRlogradouroC          = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.logradouro");
                guia.STRnumeroC              = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.numero");
                guia.STRcomplementoC         = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.complemento");
                guia.STRcodigoIBGEMunicipioC = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigoIBGEMunicipio");
                guia.STRmunicipioC           = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.municipio");
                guia.STRcodigoUFC            = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigoUF");
                guia.STRcepC                 = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.cep");
```

```
                guia.STRnumeroCNESC        = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.contratado.numeroCNES");

                guia.STRnomeProfissionalP = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.profissional.nomeProfissional");

                guia.STRsiglaConselhoP  = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.profissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoP = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.profissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoP     = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.profissional.conselhoProfissional.ufConselho");
                guia.STRcbosP           = Util.getPrimeiro(mapa,
"guiaSP_SADT.dadosSolicitante.profissional.cbos");


                guia.STRCNPJPE                    = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.identificacao.CNPJ");
                guia.STRCPFPE                     = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.identificacao.CPF");
                guia.STRcodigoPrestadorNaOperadoraPE = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.identificacao.codigoPrestadorNaOperadora");
                guia.STRnomeContratadoPE          = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.nomeContratado");

                guia.STRtipoLogradouroPE     = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.tipoLogradouro");
                guia.STRlogradouroPE         = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.logradouro");
                guia.STRnumeroPE             = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.numero");
                guia.STRcomplementoPE        = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.complemento");
                guia.STRcodigoIBGEMunicipioPE = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.codigoIBGEMunicipio");
                guia.STRmunicipioPE          = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.municipio");
                guia.STRcodigoUFPE           = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.codigoUF");
                guia.STRcepPE                = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.enderecoContratado.cep");
                guia.STRnumeroCNESPE         = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.numeroCNES");

                guia.STRnomeExecutantePEC = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.nomeExecutante");

                guia.STRsiglaConselhoPEC  = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.conselhoProfissional.siglaC
onselho");
                guia.STRnumeroConselhoPEC = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.conselhoProfissional.numer
oConselho");
                guia.STRufConselhoPEC     = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.conselhoProfissional.ufCon
selho");
                guia.STRcodigoCBOSPEC     = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.codigoCBOS");
```

```
                guia.STRcpfPEC                    = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.codigoProfissionalCompl.cp
f");
                guia.STRcodigoPrestadorNaOperadoraPEC = Util.getPrimeiro(mapa,
"guiaSP_SADT.prestadorExecutante.profissionalExecutanteCompl.codigoProfissionalCompl.co
digoPrestadorNaOperadora");


                guia.STRindicacaoClinica = Util.getPrimeiro(mapa,
"guiaSP_SADT.indicacaoClinica");


                guia.STRcaraterAtendimento = Util.getPrimeiro(mapa,
"guiaSP_SADT.caraterAtendimento");


                guia.STRdataHoraAtendimento = Util.getPrimeiro(mapa,
"guiaSP_SADT.dataHoraAtendimento");


                guia.STRnomeTabela        = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.CID.nomeTabela");
                guia.STRcodigoDiagnostico   = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.CID.codigoDiagnostico");
                guia.STRdescricaoDiagnostico = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.CID.descricaoDiagnostico");
                guia.STRtipoDoenca        = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.tipoDoenca");

                guia.STRvalor          = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.tempoReferidoEvolucaoDoenca.valor");
                guia.STRunidadeTempo     = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.tempoReferidoEvolucaoDoenca.unidadeTempo");
                guia.STRindicadorAcidente = Util.getPrimeiro(mapa,
"guiaSP_SADT.diagnosticoAtendimento.indicadorAcidente");


                guia.STRtipoSaida = Util.getPrimeiro(mapa, "guiaSP_SADT.tipoSaida");


                guia.STRtipoAtendimento = Util.getPrimeiro(mapa,
"guiaSP_SADT.tipoAtendimento");


                 guia.STRcpfPR                    = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.codigoProfiss
ional.cpf");
                 guia.STRcodigoPrestadorNaOperadoraPR = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.codigoProfiss
ional.codigoPrestadorNaOperadora");

                 guia.STRsiglaConselhoPRCP  = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.codigoProfiss
ional.conselhoProfissional.siglaConselho");
                 guia.STRnumeroConselhoPRCP = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.codigoProfiss
ional.conselhoProfissional.numeroConselho");
                 guia.STRufConselhoPRCP     = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.codigoProfiss
ional.conselhoProfissional.ufConselho");
```

```
                guia.STRnomeExecutanteID = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.identificacao
Profissional.nomeExecutante");

                guia.STRsiglaConselhoID      = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.identificacao
Profissional.conselhoProfissional.siglaConselho");
                guia.STRnumeroConselhoID     = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.identificacao
Profissional.conselhoProfissional.numeroConselho");
                guia.STRufConselhoID         = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.identificacao
Profissional.conselhoProfissional.ufConselho");
                guia.STRcodigoCBOSID         = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.identificacao
Profissional.codigoCBOS");
                guia.STRcpfID                = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.cpf");
                guia.STRposicaoProfissionalID = Util.getPrimeiro(mapa,
"guiaSP_SADT.procedimentosRealizados.procedimentos.equipe.membroEquipe.posicaoProfis
sional");

                guia.STRcodigoPR             =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.procedimento.codigo");
                guia.STRtipoTabelaPR         =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.procedimento.tipoTabela")
;
                guia.STRdescricaoPR          =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.procedimento.descricao");
                guia.STRdataPR               =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.data");
                guia.STRhoraInicioPR         =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.horaInicio");
                guia.STRhoraFimPR            =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.horaFim");
                guia.STRquantidadeRealizadaPR =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.quantidadeRealizada");
                guia.STRviaAcessoPR          =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.viaAcesso");
                guia.STRtecnicaUtilizadaPR   =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.tecnicaUtilizada");
                guia.STRreducaoAcrescimoPR   =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.reducaoAcrescimo");
                guia.STRvalorPR              =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.valor");
                guia.STRvalorTotalPR         =
mapa.get("guiaSP_SADT.procedimentosRealizados.procedimentos.valorTotal");


                guia.STRcodigoOD             = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.identificadorDespesa.codigo");
                guia.STRtipoTabelaOD         = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.identificadorDespesa.tipoTabela");
                guia.STRdescricaoOD          = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.identificadorDespesa.descricao");
                guia.STRtipoDespesaOD        = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.tipoDespesa");
                guia.STRdataRealizacaoOD     = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.dataRealizacao");
```

```
                guia.STRhoraInicialOD          = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.horaInicial");
                guia.STRhoraFinalOD            = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.horaFinal");
                guia.STRreducaoAcrescimoOD     = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.reducaoAcrescimo");
                guia.STRquantidadeOD           = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.quantidade");
                guia.STRvalorUnitarioOD        = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.valorUnitario");
                guia.STRvalorTotalOD           = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.despesa.valorTotal");
                guia.STRtotalGeralOutrasDespesasOD = Util.getPrimeiro(mapa,
"guiaSP_SADT.outrasDespesas.totalGeralOutrasDespesas");

                guia.STRcodigoOPM      =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.OPM.codigo");
                guia.STRtipoTabelaOPM  =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.OPM.tipoTabela");
                guia.STRdescricaoOPM   =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.OPM.descricao");
                guia.STRquantidadeOPM  =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.quantidade");
                guia.STRcodigoBarraOPM =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.codigoBarra");
                guia.STRvalorUnitarioOPM =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.valorUnitario");
                guia.STRvalorTotalOPM  =
mapa.get("guiaSP_SADT.OPMUtilizada.OPM.identificacaoOPM.valorTotal");

                guia.STRvalorTotalOPMOPM = Util.getPrimeiro(mapa,
"guiaSP_SADT.OPMUtilizada.valorTotalOPM");


                guia.STRservicosExecutados = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.servicosExecutados");
                guia.STRdiarias        = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.diarias");
                guia.STRtaxas          = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.taxas");
                guia.STRmateriais      = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.materiais");
                guia.STRmedicamentos   = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.medicamentos");
                guia.STRgases          = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.gases");
                guia.STRtotalGeral     = Util.getPrimeiro(mapa,
"guiaSP_SADT.valorTotal.totalGeral");

                guia.STRobservacao = Util.getPrimeiro(mapa, "guiaSP_SADT.observacao");

                return guia;
        }


        public String getSTRregistroANS() {
                return STRregistroANS;
        }
        public void setSTRregistroANS(String rregistroANS) {
                STRregistroANS = rregistroANS;
```

```
}
public String getSTRdataEmissaoGuia() {
        return STRdataEmissaoGuia;
}
public void setSTRdataEmissaoGuia(String rdataEmissaoGuia) {
        STRdataEmissaoGuia = rdataEmissaoGuia;
}
public String getSTRnumeroGuiaPrestador() {
        return STRnumeroGuiaPrestador;
}
public void setSTRnumeroGuiaPrestador(String rnumeroGuiaPrestador) {
        STRnumeroGuiaPrestador = rnumeroGuiaPrestador;
}
public String getSTRnumeroGuiaOperadora() {
        return STRnumeroGuiaOperadora;
}
public void setSTRnumeroGuiaOperadora(String rnumeroGuiaOperadora) {
        STRnumeroGuiaOperadora = rnumeroGuiaOperadora;
}
public String getSTRnumeroGuiaPrincipal() {
        return STRnumeroGuiaPrincipal;
}
public void setSTRnumeroGuiaPrincipal(String rnumeroGuiaPrincipal) {
        STRnumeroGuiaPrincipal = rnumeroGuiaPrincipal;
}
public String getSTRdataAutorizacao() {
        return STRdataAutorizacao;
}
public void setSTRdataAutorizacao(String rdataAutorizacao) {
        STRdataAutorizacao = rdataAutorizacao;
}
public String getSTRsenhaAutorizacao() {
        return STRsenhaAutorizacao;
}
public void setSTRsenhaAutorizacao(String rsenhaAutorizacao) {
        STRsenhaAutorizacao = rsenhaAutorizacao;
}
public String getSTRvalidadeSenha() {
        return STRvalidadeSenha;
}
public void setSTRvalidadeSenha(String rvalidadeSenha) {
        STRvalidadeSenha = rvalidadeSenha;
}
public String getSTRnumeroCarteira() {
        return STRnumeroCarteira;
}
public void setSTRnumeroCarteira(String rnumeroCarteira) {
        STRnumeroCarteira = rnumeroCarteira;
}
public String getSTRnomeBeneficiario() {
        return STRnomeBeneficiario;
}
public void setSTRnomeBeneficiario(String rnomeBeneficiario) {
        STRnomeBeneficiario = rnomeBeneficiario;
}
public String getSTRnomePlano() {
        return STRnomePlano;
}
public void setSTRnomePlano(String rnomePlano) {
        STRnomePlano = rnomePlano;
```

```
}
public String getSTRvalidadeCarteira() {
        return STRvalidadeCarteira;
}
public void setSTRvalidadeCarteira(String rvalidadeCarteira) {
        STRvalidadeCarteira = rvalidadeCarteira;
}
public String getSTRnumeroCNS() {
        return STRnumeroCNS;
}
public void setSTRnumeroCNS(String rnumeroCNS) {
        STRnumeroCNS = rnumeroCNS;
}
public String getSTRidentificadorBeneficiario() {
        return STRidentificadorBeneficiario;
}
public void setSTRidentificadorBeneficiario(String ridentificadorBeneficiario) {
        STRidentificadorBeneficiario = ridentificadorBeneficiario;
}
public String getSTRCNPJC() {
        return STRCNPJC;
}
public void setSTRCNPJC(String strcnpjc) {
        STRCNPJC = strcnpjc;
}
public String getSTRcpfC() {
        return STRcpfC;
}
public void setSTRcpfC(String rcpfC) {
        STRcpfC = rcpfC;
}
public String getSTRcodigoPrestadorNaOperadoraC() {
        return STRcodigoPrestadorNaOperadoraC;
}
public void setSTRcodigoPrestadorNaOperadoraC(
                String rcodigoPrestadorNaOperadoraC) {
        STRcodigoPrestadorNaOperadoraC = rcodigoPrestadorNaOperadoraC;
}
public String getSTRsiglaConselhoC() {
        return STRsiglaConselhoC;
}
public void setSTRsiglaConselhoC(String rsiglaConselhoC) {
        STRsiglaConselhoC = rsiglaConselhoC;
}
public String getSTRnumeroConselhoC() {
        return STRnumeroConselhoC;
}
public void setSTRnumeroConselhoC(String rnumeroConselhoC) {
        STRnumeroConselhoC = rnumeroConselhoC;
}
public String getSTRufConselhoC() {
        return STRufConselhoC;
}
public void setSTRufConselhoC(String rufConselhoC) {
        STRufConselhoC = rufConselhoC;
}
public String getSTRnomeContratadoC() {
        return STRnomeContratadoC;
}
public void setSTRnomeContratadoC(String rnomeContratadoC) {
```

```
        STRnomeContratadoC = rnomeContratadoC;
}
public String getSTRntipoLogradouroC() {
        return STRntipoLogradouroC;
}
public void setSTRntipoLogradouroC(String rntipoLogradouroC) {
        STRntipoLogradouroC = rntipoLogradouroC;
}
public String getSTRlogradouroC() {
        return STRlogradouroC;
}
public void setSTRlogradouroC(String rlogradouroC) {
        STRlogradouroC = rlogradouroC;
}
public String getSTRnumeroC() {
        return STRnumeroC;
}
public void setSTRnumeroC(String rnumeroC) {
        STRnumeroC = rnumeroC;
}
public String getSTRcomplementoC() {
        return STRcomplementoC;
}
public void setSTRcomplementoC(String rcomplementoC) {
        STRcomplementoC = rcomplementoC;
}
public String getSTRcodigoIBGEMunicipioC() {
        return STRcodigoIBGEMunicipioC;
}
public void setSTRcodigoIBGEMunicipioC(String rcodigoIBGEMunicipioC) {
        STRcodigoIBGEMunicipioC = rcodigoIBGEMunicipioC;
}
public String getSTRmunicipioC() {
        return STRmunicipioC;
}
public void setSTRmunicipioC(String rmunicipioC) {
        STRmunicipioC = rmunicipioC;
}
public String getSTRcodigoUFC() {
        return STRcodigoUFC;
}
public void setSTRcodigoUFC(String rcodigoUFC) {
        STRcodigoUFC = rcodigoUFC;
}
public String getSTRcepC() {
        return STRcepC;
}
public void setSTRcepC(String rcepC) {
        STRcepC = rcepC;
}
public String getSTRnumeroCNESC() {
        return STRnumeroCNESC;
}
public void setSTRnumeroCNESC(String rnumeroCNESC) {
        STRnumeroCNESC = rnumeroCNESC;
}
public String getSTRnomeProfissionalP() {
        return STRnomeProfissionalP;
}
public void setSTRnomeProfissionalP(String rnomeProfissionalP) {
```

```
        STRnomeProfissionalP = rnomeProfissionalP;
}
public String getSTRsiglaConselhoP() {
        return STRsiglaConselhoP;
}
public void setSTRsiglaConselhoP(String rsiglaConselhoP) {
        STRsiglaConselhoP = rsiglaConselhoP;
}
public String getSTRnumeroConselhoP() {
        return STRnumeroConselhoP;
}
public void setSTRnumeroConselhoP(String rnumeroConselhoP) {
        STRnumeroConselhoP = rnumeroConselhoP;
}
public String getSTRufConselhoP() {
        return STRufConselhoP;
}
public void setSTRufConselhoP(String rufConselhoP) {
        STRufConselhoP = rufConselhoP;
}
public String getSTRcbosP() {
        return STRcbosP;
}
public void setSTRcbosP(String rcbosP) {
        STRcbosP = rcbosP;
}
public String getSTRCNPJPE() {
        return STRCNPJPE;
}
public void setSTRCNPJPE(String strcnpjpe) {
        STRCNPJPE = strcnpjpe;
}
public String getSTRCPFPE() {
        return STRCPFPE;
}
public void setSTRCPFPE(String strcpfpe) {
        STRCPFPE = strcpfpe;
}
public String getSTRcodigoPrestadorNaOperadoraPE() {
        return STRcodigoPrestadorNaOperadoraPE;
}
public void setSTRcodigoPrestadorNaOperadoraPE(
                String rcodigoPrestadorNaOperadoraPE) {
        STRcodigoPrestadorNaOperadoraPE = rcodigoPrestadorNaOperadoraPE;
}
public String getSTRnomeContratadoPE() {
        return STRnomeContratadoPE;
}
public void setSTRnomeContratadoPE(String rnomeContratadoPE) {
        STRnomeContratadoPE = rnomeContratadoPE;
}
public String getSTRtipoLogradouroPE() {
        return STRtipoLogradouroPE;
}
public void setSTRtipoLogradouroPE(String rtipoLogradouroPE) {
        STRtipoLogradouroPE = rtipoLogradouroPE;
}
public String getSTRlogradouroPE() {
        return STRlogradouroPE;
}
```

```
public void setSTRlogradouroPE(String rlogradouroPE) {
        STRlogradouroPE = rlogradouroPE;
}
public String getSTRnumeroPE() {
        return STRnumeroPE;
}
public void setSTRnumeroPE(String rnumeroPE) {
        STRnumeroPE = rnumeroPE;
}
public String getSTRcomplementoPE() {
        return STRcomplementoPE;
}
public void setSTRcomplementoPE(String rcomplementoPE) {
        STRcomplementoPE = rcomplementoPE;
}
public String getSTRcodigoIBGEMunicipioPE() {
        return STRcodigoIBGEMunicipioPE;
}
public void setSTRcodigoIBGEMunicipioPE(String rcodigoIBGEMunicipioPE) {
        STRcodigoIBGEMunicipioPE = rcodigoIBGEMunicipioPE;
}
public String getSTRmunicipioPE() {
        return STRmunicipioPE;
}
public void setSTRmunicipioPE(String rmunicipioPE) {
        STRmunicipioPE = rmunicipioPE;
}
public String getSTRcodigoUFPE() {
        return STRcodigoUFPE;
}
public void setSTRcodigoUFPE(String rcodigoUFPE) {
        STRcodigoUFPE = rcodigoUFPE;
}
public String getSTRcepPE() {
        return STRcepPE;
}
public void setSTRcepPE(String rcepPE) {
        STRcepPE = rcepPE;
}
public String getSTRnumeroCNESPE() {
        return STRnumeroCNESPE;
}
public void setSTRnumeroCNESPE(String rnumeroCNESPE) {
        STRnumeroCNESPE = rnumeroCNESPE;
}
public String getSTRnomeExecutantePEC() {
        return STRnomeExecutantePEC;
}
public void setSTRnomeExecutantePEC(String rnomeExecutantePEC) {
        STRnomeExecutantePEC = rnomeExecutantePEC;
}
public String getSTRsiglaConselhoPEC() {
        return STRsiglaConselhoPEC;
}
public void setSTRsiglaConselhoPEC(String rsiglaConselhoPEC) {
        STRsiglaConselhoPEC = rsiglaConselhoPEC;
}
public String getSTRnumeroConselhoPEC() {
        return STRnumeroConselhoPEC;
}
```

```
public void setSTRnumeroConselhoPEC(String rnumeroConselhoPEC) {
        STRnumeroConselhoPEC = rnumeroConselhoPEC;
}
public String getSTRufConselhoPEC() {
        return STRufConselhoPEC;
}
public void setSTRufConselhoPEC(String rufConselhoPEC) {
        STRufConselhoPEC = rufConselhoPEC;
}
public String getSTRcodigoCBOSPEC() {
        return STRcodigoCBOSPEC;
}
public void setSTRcodigoCBOSPEC(String rcodigoCBOSPEC) {
        STRcodigoCBOSPEC = rcodigoCBOSPEC;
}
public String getSTRcpfPEC() {
        return STRcpfPEC;
}
public void setSTRcpfPEC(String rcpfPEC) {
        STRcpfPEC = rcpfPEC;
}
public String getSTRcodigoPrestadorNaOperadoraPEC() {
        return STRcodigoPrestadorNaOperadoraPEC;
}
public void setSTRcodigoPrestadorNaOperadoraPEC(
                String rcodigoPrestadorNaOperadoraPEC) {
        STRcodigoPrestadorNaOperadoraPEC = rcodigoPrestadorNaOperadoraPEC;
}
public String getSTRindicacaoClinica() {
        return STRindicacaoClinica;
}
public void setSTRindicacaoClinica(String rindicacaoClinica) {
        STRindicacaoClinica = rindicacaoClinica;
}
public String getSTRcaraterAtendimento() {
        return STRcaraterAtendimento;
}
public void setSTRcaraterAtendimento(String rcaraterAtendimento) {
        STRcaraterAtendimento = rcaraterAtendimento;
}
public String getSTRdataHoraAtendimento() {
        return STRdataHoraAtendimento;
}
public void setSTRdataHoraAtendimento(String rdataHoraAtendimento) {
        STRdataHoraAtendimento = rdataHoraAtendimento;
}
public String getSTRnomeTabela() {
        return STRnomeTabela;
}
public void setSTRnomeTabela(String rnomeTabela) {
        STRnomeTabela = rnomeTabela;
}
public String getSTRcodigoDiagnostico() {
        return STRcodigoDiagnostico;
}
public void setSTRcodigoDiagnostico(String rcodigoDiagnostico) {
        STRcodigoDiagnostico = rcodigoDiagnostico;
}
public String getSTRdescricaoDiagnostico() {
        return STRdescricaoDiagnostico;
```

```
}
public void setSTRdescricaoDiagnostico(String rdescricaoDiagnostico) {
        STRdescricaoDiagnostico = rdescricaoDiagnostico;
}
public String getSTRtipoDoenca() {
        return STRtipoDoenca;
}
public void setSTRtipoDoenca(String rtipoDoenca) {
        STRtipoDoenca = rtipoDoenca;
}
public String getSTRvalor() {
        return STRvalor;
}
public void setSTRvalor(String rvalor) {
        STRvalor = rvalor;
}
public String getSTRunidadeTempo() {
        return STRunidadeTempo;
}
public void setSTRunidadeTempo(String runidadeTempo) {
        STRunidadeTempo = runidadeTempo;
}
public String getSTRindicadorAcidente() {
        return STRindicadorAcidente;
}
public void setSTRindicadorAcidente(String rindicadorAcidente) {
        STRindicadorAcidente = rindicadorAcidente;
}
public String getSTRtipoSaida() {
        return STRtipoSaida;
}
public void setSTRtipoSaida(String rtipoSaida) {
        STRtipoSaida = rtipoSaida;
}
public String getSTRtipoAtendimento() {
        return STRtipoAtendimento;
}
public void setSTRtipoAtendimento(String rtipoAtendimento) {
        STRtipoAtendimento = rtipoAtendimento;
}
public String getSTRcpfPR() {
        return STRcpfPR;
}
public void setSTRcpfPR(String rcpfPR) {
        STRcpfPR = rcpfPR;
}
public String getSTRcodigoPrestadorNaOperadoraPR() {
        return STRcodigoPrestadorNaOperadoraPR;
}
public void setSTRcodigoPrestadorNaOperadoraPR(
                String rcodigoPrestadorNaOperadoraPR) {
        STRcodigoPrestadorNaOperadoraPR = rcodigoPrestadorNaOperadoraPR;
}
public String getSTRsiglaConselhoPRCP() {
        return STRsiglaConselhoPRCP;
}
public void setSTRsiglaConselhoPRCP(String rsiglaConselhoPRCP) {
        STRsiglaConselhoPRCP = rsiglaConselhoPRCP;
}
public String getSTRnumeroConselhoPRCP() {
```

```
        return STRnumeroConselhoPRCP;
}
public void setSTRnumeroConselhoPRCP(String rnumeroConselhoPRCP) {
        STRnumeroConselhoPRCP = rnumeroConselhoPRCP;
}
public String getSTRufConselhoPRCP() {
        return STRufConselhoPRCP;
}
public void setSTRufConselhoPRCP(String rufConselhoPRCP) {
        STRufConselhoPRCP = rufConselhoPRCP;
}
public String getSTRnomeExecutanteID() {
        return STRnomeExecutanteID;
}
public void setSTRnomeExecutanteID(String rnomeExecutanteID) {
        STRnomeExecutanteID = rnomeExecutanteID;
}
public String getSTRsiglaConselhoID() {
        return STRsiglaConselhoID;
}
public void setSTRsiglaConselhoID(String rsiglaConselhoID) {
        STRsiglaConselhoID = rsiglaConselhoID;
}
public String getSTRnumeroConselhoID() {
        return STRnumeroConselhoID;
}
public void setSTRnumeroConselhoID(String rnumeroConselhoID) {
        STRnumeroConselhoID = rnumeroConselhoID;
}
public String getSTRufConselhoID() {
        return STRufConselhoID;
}
public void setSTRufConselhoID(String rufConselhoID) {
        STRufConselhoID = rufConselhoID;
}
public String getSTRcodigoCBOSID() {
        return STRcodigoCBOSID;
}
public void setSTRcodigoCBOSID(String rcodigoCBOSID) {
        STRcodigoCBOSID = rcodigoCBOSID;
}
public String getSTRcpfID() {
        return STRcpfID;
}
public void setSTRcpfID(String rcpfID) {
        STRcpfID = rcpfID;
}
public String getSTRposicaoProfissionalID() {
        return STRposicaoProfissionalID;
}
public void setSTRposicaoProfissionalID(String rposicaoProfissionalID) {
        STRposicaoProfissionalID = rposicaoProfissionalID;
}
public String[] getSTRcodigoPR() {
        return STRcodigoPR;
}
public void setSTRcodigoPR(String[] rcodigoPR) {
        STRcodigoPR = rcodigoPR;
}
public String[] getSTRtipoTabelaPR() {
```

```
        return STRtipoTabelaPR;
}
public void setSTRtipoTabelaPR(String[] rtipoTabelaPR) {
        STRtipoTabelaPR = rtipoTabelaPR;
}
public String[] getSTRdescricaoPR() {
        return STRdescricaoPR;
}
public void setSTRdescricaoPR(String[] rdescricaoPR) {
        STRdescricaoPR = rdescricaoPR;
}
public String[] getSTRdataPR() {
        return STRdataPR;
}
public void setSTRdataPR(String[] rdataPR) {
        STRdataPR = rdataPR;
}
public String[] getSTRhoraInicioPR() {
        return STRhoraInicioPR;
}
public void setSTRhoraInicioPR(String[] rhoraInicioPR) {
        STRhoraInicioPR = rhoraInicioPR;
}
public String[] getSTRhoraFimPR() {
        return STRhoraFimPR;
}
public void setSTRhoraFimPR(String[] rhoraFimPR) {
        STRhoraFimPR = rhoraFimPR;
}
public String[] getSTRquantidadeRealizadaPR() {
        return STRquantidadeRealizadaPR;
}
public void setSTRquantidadeRealizadaPR(String[] rquantidadeRealizadaPR) {
        STRquantidadeRealizadaPR = rquantidadeRealizadaPR;
}
public String[] getSTRviaAcessoPR() {
        return STRviaAcessoPR;
}
public void setSTRviaAcessoPR(String[] rviaAcessoPR) {
        STRviaAcessoPR = rviaAcessoPR;
}
public String[] getSTRtecnicaUtilizadaPR() {
        return STRtecnicaUtilizadaPR;
}
public void setSTRtecnicaUtilizadaPR(String[] rtecnicaUtilizadaPR) {
        STRtecnicaUtilizadaPR = rtecnicaUtilizadaPR;
}
public String[] getSTRreducaoAcrescimoPR() {
        return STRreducaoAcrescimoPR;
}
public void setSTRreducaoAcrescimoPR(String[] rreducaoAcrescimoPR) {
        STRreducaoAcrescimoPR = rreducaoAcrescimoPR;
}
public String[] getSTRvalorPR() {
        return STRvalorPR;
}
public void setSTRvalorPR(String[] rvalorPR) {
        STRvalorPR = rvalorPR;
}
public String[] getSTRvalorTotalPR() {
```

```
        return STRvalorTotalPR;
}
public void setSTRvalorTotalPR(String[] rvalorTotalPR) {
        STRvalorTotalPR = rvalorTotalPR;
}
public String getSTRcodigoOD() {
        return STRcodigoOD;
}
public void setSTRcodigoOD(String rcodigoOD) {
        STRcodigoOD = rcodigoOD;
}
public String getSTRtipoTabelaOD() {
        return STRtipoTabelaOD;
}
public void setSTRtipoTabelaOD(String rtipoTabelaOD) {
        STRtipoTabelaOD = rtipoTabelaOD;
}
public String getSTRdescricaoOD() {
        return STRdescricaoOD;
}
public void setSTRdescricaoOD(String rdescricaoOD) {
        STRdescricaoOD = rdescricaoOD;
}
public String getSTRtipoDespesaOD(){
        return STRtipoDespesaOD;
}
public void setSTRtipoDespesaOD(String rtipoDespesaOD) {
        STRtipoDespesaOD = rtipoDespesaOD;
}
public String getSTRdataRealizacaoOD() {
        return STRdataRealizacaoOD;
}
public void setSTRdataRealizacaoOD(String rdataRealizacaoOD) {
        STRdataRealizacaoOD = rdataRealizacaoOD;
}
public String getSTRhoraInicialOD() {
        return STRhoraInicialOD;
}
public void setSTRhoraInicialOD(String rhoraInicialOD) {
        STRhoraInicialOD = rhoraInicialOD;
}
public String getSTRhoraFinalOD() {
        return STRhoraFinalOD;
}
public void setSTRhoraFinalOD(String rhoraFinalOD) {
        STRhoraFinalOD = rhoraFinalOD;
}
public String getSTRreducaoAcrescimoOD() {
        return STRreducaoAcrescimoOD;
}
public void setSTRreducaoAcrescimoOD(String rreducaoAcrescimoOD) {
        STRreducaoAcrescimoOD = rreducaoAcrescimoOD;
}
public String getSTRquantidadeOD() {
        return STRquantidadeOD;
}
public void setSTRquantidadeOD(String rquantidadeOD) {
        STRquantidadeOD = rquantidadeOD;
}
public String getSTRvalorUnitarioOD() {
```

```
            return STRvalorUnitarioOD;
        }
        public void setSTRvalorUnitarioOD(String rvalorUnitarioOD) {
            STRvalorUnitarioOD = rvalorUnitarioOD;
        }
        public String getSTRvalorTotalOD() {
            return STRvalorTotalOD;
        }
        public void setSTRvalorTotalOD(String rvalorTotalOD) {
            STRvalorTotalOD = rvalorTotalOD;
        }
        public String getSTRtotalGeralOutrasDespesasOD() {
            return STRtotalGeralOutrasDespesasOD;
        }
        public void setSTRtotalGeralOutrasDespesasOD(String
rtotalGeralOutrasDespesasOD) {
            STRtotalGeralOutrasDespesasOD = rtotalGeralOutrasDespesasOD;
        }
        public String[] getSTRcodigoOPM() {
            return STRcodigoOPM;
        }
        public void setSTRcodigoOPM(String[] rcodigoOPM) {
            STRcodigoOPM = rcodigoOPM;
        }
        public String[] getSTRtipoTabelaOPM() {
            return STRtipoTabelaOPM;
        }
        public void setSTRtipoTabelaOPM(String[] rtipoTabelaOPM) {
            STRtipoTabelaOPM = rtipoTabelaOPM;
        }
        public String[] getSTRdescricaoOPM() {
            return STRdescricaoOPM;
        }
        public void setSTRdescricaoOPM(String[] rdescricaoOPM) {
            STRdescricaoOPM = rdescricaoOPM;
        }
        public String[] getSTRquantidadeOPM() {
            return STRquantidadeOPM;
        }
        public void setSTRquantidadeOPM(String[] rquantidadeOPM) {
            STRquantidadeOPM = rquantidadeOPM;
        }
        public String[] getSTRcodigoBarraOPM() {
            return STRcodigoBarraOPM;
        }
        public void setSTRcodigoBarraOPM(String[] rcodigoBarraOPM) {
            STRcodigoBarraOPM = rcodigoBarraOPM;
        }
        public String[] getSTRvalorUnitarioOPM() {
            return STRvalorUnitarioOPM;
        }
        public void setSTRvalorUnitarioOPM(String[] rvalorUnitarioOPM) {
            STRvalorUnitarioOPM = rvalorUnitarioOPM;
        }
        public String[] getSTRvalorTotalOPM() {
            return STRvalorTotalOPM;
        }
        public void setSTRvalorTotalOPM(String[] rvalorTotalOPM) {
            STRvalorTotalOPM = rvalorTotalOPM;
        }
```

```
	public String getSTRvalorTotalOPMOPM() {
		return STRvalorTotalOPMOPM;
	}
	public void setSTRvalorTotalOPMOPM(String rvalorTotalOPMOPM) {
		STRvalorTotalOPMOPM = rvalorTotalOPMOPM;
	}
	public String getSTRservicosExecutados() {
		return STRservicosExecutados;
	}
	public void setSTRservicosExecutados(String rservicosExecutados) {
		STRservicosExecutados = rservicosExecutados;
	}
	public String getSTRdiarias() {
		return STRdiarias;
	}
	public void setSTRdiarias(String rdiarias) {
		STRdiarias = rdiarias;
	}
	public String getSTRtaxas() {
		return STRtaxas;
	}
	public void setSTRtaxas(String rtaxas) {
		STRtaxas = rtaxas;
	}
	public String getSTRmateriais() {
		return STRmateriais;
	}
	public void setSTRmateriais(String rmateriais) {
		STRmateriais = rmateriais;
	}
	public String getSTRmedicamentos() {
		return STRmedicamentos;
	}
	public void setSTRmedicamentos(String rmedicamentos) {
		STRmedicamentos = rmedicamentos;
	}
	public String getSTRgases() {
		return STRgases;
	}
	public void setSTRgases(String rgases) {
		STRgases = rgases;
	}
	public String getSTRtotalGeral() {
		return STRtotalGeral;
	}
	public void setSTRtotalGeral(String rtotalGeral) {
		STRtotalGeral = rtotalGeral;
	}
	public String getSTRobservacao() {
		return STRobservacao;
	}
	public void setSTRobservacao(String robservacao) {
		STRobservacao = robservacao;
	}
}
```

Nome do arquivo: ClLote.java

```
package com.classes;
```

```
import com.DAO.DAOLote;

public class ClLote {
	private Long inicioLote;
	private Long fimLote;
	private Long proximo;
	private Integer codigoOperadora;
	private Integer idLote;

	public Long getInicioLote() {
		return inicioLote;
	}
	public void setInicioLote(Long inicioLote) {
		this.inicioLote = inicioLote;
	}
	public Long getFimLote() {
		return fimLote;
	}
	public void setFimLote(Long fimLote) {
		this.fimLote = fimLote;
	}
	public Long getProximo() {
		return proximo;
	}
	public void setProximo(Long proximo) {
		this.proximo = proximo;
	}
	public Integer getCodigoOperadora() {
		return codigoOperadora;
	}
	public void setCodigoOperadora(Integer codigoOperadora) {
		this.codigoOperadora = codigoOperadora;
	}
	public Integer getIdLote() {
		return idLote;
	}
	public void setIdLote(Integer idLote){
		this.idLote = idLote;
	}

	public void cadastraLote(ClLote lote){
		DAOLote dAOLote = new DAOLote();
		dAOLote.cadastraLote(lote);
	}
	public Long getProximoNumeroLote(Integer codigoOper){
		DAOLote dAOLote = new DAOLote();
		Long proximoNumeroGuia = dAOLote.getProximoNumeroLote(codigoOper);
		return proximoNumeroGuia;
	}

}
```

Nome do arquivo: ClOperadora.java

```
package com.classes;

import java.util.ArrayList;
import java.util.List;
```

```
import com.DAO.DAOOperadora;

public class ClOperadora {
	private String registro;
	private String nome;
	private Integer codigo;
	private String cnpj;
	private String idPrestador = "0";

	public String getIdPrestador() {
		return idPrestador;
	}
	public void setIdPrestador(String idPrestador) {
		this.idPrestador = idPrestador;
	}
	public String getCnpj() {
		return cnpj;
	}
	public void setCnpj(String cnpj) {
		this.cnpj = cnpj;
	}
	public void setRegistro(String registro) {
		this.registro = registro;
	}
	public String getRegistro() {
		return registro;
	}
	public void setNome(String nome) {
		this.nome = nome;
	}
	public String getNome() {
		return nome;
	}
	public void setCodigo(Integer codigo) {
		this.codigo = codigo;
	}
	public Integer getCodigo() {
		return codigo;
	}

	public void removerOperadora(int codigo){
		DAOOperadora daooperadora = new DAOOperadora();
		daooperadora.removerOperadora(codigo);
	}
	public void cadastrarOperadora(ClOperadora operadora){
		DAOOperadora daooperadora = new DAOOperadora();
		daooperadora.cadastrarOperadora(operadora);
	}
	public ClOperadora obterDetalhesOperadora (int codigo){
		DAOOperadora daooperadora = new DAOOperadora();
		ClOperadora clOperadora  = new ClOperadora();
		clOperadora          = daooperadora.obterDetalhesOperadora(codigo);
		return clOperadora;
	}
	public void editarOperadora(ClOperadora operadora){
		DAOOperadora daooperadora = new DAOOperadora();
		daooperadora.editarOperadora(operadora);
	}
	public void editarCamposObrigatorios(Integer codigoOperadora, List<String> campos){
```

```
                DAOOperadora daooperadora = new DAOOperadora();
                daooperadora.editarCamposObrigatorios(codigoOperadora, campos);
        }
        public List<String> listaCampos(Integer codigoOperadora){
                DAOOperadora daooperadora = new DAOOperadora();
                List<String> campos      = new ArrayList<String>();
                campos                   = daooperadora.listaCampos(codigoOperadora);
                return campos;
        }
}
```

Nome do arquivo: ClPreFisico.java

```
package com.classes;

import com.DAO.DAOPrestador;

public class ClPreFisico extends ClPrestador{
        private String cpf;
        private String conselhoprof;
        private String numconselho;
        private String ufconselho;
        private String cbos;

        public void setCpf(String cpf) {
                this.cpf = cpf;
        }
        public String getCpf() {
                return cpf;
        }
        public void setConselhoprof(String conselhoprof) {
                this.conselhoprof = conselhoprof;
        }
        public String getConselhoprof() {
                return conselhoprof;
        }
        public void setUfconselho(String ufconselho) {
                this.ufconselho = ufconselho;
        }
        public String getUfconselho() {
                return ufconselho;
        }
        public void setCbos(String cbos) {
                this.cbos = cbos;
        }
        public String getCbos() {
                return cbos;
        }
        public void setNumconselho(String numconselho) {
                this.numconselho = numconselho;
        }
        public String getNumconselho() {
                return numconselho;
        }
        public void cadastrarPreFisico(ClPreFisico prefisico){
                DAOPrestador daoprefisico = new DAOPrestador();
                daoprefisico.cadastrarPreFisico(prefisico);
        }
        public void editarPreFisico(ClPreFisico prefisico){
```

```
                DAOPrestador daoprefisico = new DAOPrestador();
                daoprefisico.editarPreFisico(prefisico);
        }
        public ClPreFisico obterPreFisicoDetalhes(String cpf) {
                ClPreFisico preFisico    = new ClPreFisico();
                DAOPrestador daoprefisico = new DAOPrestador();
                preFisico                = daoprefisico.obterPreFisicoDetalhes(cpf);
                return preFisico;
        }

}
```

Nome do arquivo: ClPreJuridico.java

```
package com.classes;

import com.DAO.DAOPrestador;

public class ClPreJuridico extends ClPrestador{
        private String cnpj;

        public String getCnpj() {
                return cnpj;
        }
        public void setCnpj(String cnpj) {
                this.cnpj = cnpj;
        }

        public void cadastrarPreJuridico(ClPreJuridico prejuridico){
                DAOPrestador daoprejuridico = new DAOPrestador();
                daoprejuridico.cadastrarPreJuridico(prejuridico);
        }
        public void editarPreJuridico(ClPreJuridico prejuridico) {
                DAOPrestador daoprejuridico = new DAOPrestador();
                daoprejuridico.editarPreJuridico(prejuridico);
        }
        public ClPreJuridico obterPreJuridicoDetalhes(String cnpj) {
                DAOPrestador daoprejuridico = new DAOPrestador();
                ClPreJuridico prejuridico   = new ClPreJuridico();
                prejuridico                 = daoprejuridico.obterPreJuridicoDetalhes(cnpj);
                return prejuridico;
        }
}
```

Nome do arquivo: ClPrestador.java

```
package com.classes;

import com.DAO.DAOPrestador;

public class ClPrestador {
        private String nome;
        private String cnes;
        private String codoperadora;
        private String tipo;
        private ClEndereco endereco;

        public String getNome() {
```

```
                return nome;
        }
        public void setNome(String nome) {
                this.nome = nome;
        }
        public String getCnes() {
                return cnes;
        }
        public void setCnes(String cnes) {
                this.cnes = cnes;
        }
        public String getCodoperadora() {
                return codoperadora;
        }
        public void setCodoperadora(String cooperadora) {
                this.codoperadora = cooperadora;
        }
        public ClEndereco getEndereco() {
                return endereco;
        }
        public void setEndereco(ClEndereco endereco) {
                this.endereco = endereco;
        }
        public void setTipo(String tipo) {
                this.tipo = tipo;
        }
        public String getTipo() {
                return tipo;
        }

        public void removerPrestador(String cpfcnpj) {
                DAOPrestador daoprestador = new DAOPrestador();
                daoprestador.removerPrestador(cpfcnpj);
        }
}
```

Nome do arquivo: ClTipoLogradouro.java

```
package com.classes;

public class ClTipoLogradouro {
        private int codigo;
        private String descricao;

        public void setCodigoLogradouro(int codigo){
                this.codigo = codigo;
        }
        public void setDescricaoLogradouro(String descricao){
                this.descricao = descricao;
        }
        public int getCodigoLogradouro(){
                return this.codigo;
        }
        public String getDescricaoLogradouro(){
                return this.descricao;
        }
}
```

Nome do arquivo: CITipoPlano.java

```
package com.classes;

public class CITipoPlano {
	private int codigo;
	private String plano;

	public void setUF(int Codigo){
		this.codigo = Codigo;
	}
	public int getCodigo(){
		return this.codigo;
	}
	public void setPlano(String Plano){
		this.plano = Plano;
	}
	public String getPlano(){
		return this.plano;
	}
}
```

Nome do arquivo: CIUF.java

```
package com.classes;

public class CIUF {
	private String uf;
	private String estado;

	public void setUF(String UF){
		this.uf = UF;
	}
	public String getUF(){
		return this.uf;
	}
	public void setEstado(String Estado){
		this.estado = Estado;
	}
	public String getEstado(){
		return this.estado;
	}
}
```

Nome do arquivo: CIUsuario.java

```
package com.classes;

import com.DAO.DAOUsuario;

public class CIUsuario {
	private Integer codigo;
	private String usuario;
	private String senha;
	private String ativo;
	private String login;

	public String getLogin() {
```

```
        return login;
    }
    public void setLogin(String login) {
        this.login = login;
    }
    public ClUsuario(){
        this.usuario = "";
        this.senha = "";
        this.ativo = "";
    }
    public void setCodigo(Integer c){
        this.codigo = c;
    }
    public void setUsuario(String u){
        this.usuario = u;
    }
    public void setAtivo(String a){
        this.ativo = a;
    }
    public void setSenha(String s){
        this.senha = s;
    }
    public Integer getCodigo(){
        return this.codigo;
    }
    public String getUsuario(){
        return this.usuario;
    }
    public String getSenha(){
        return this.senha;
    }
    public String getAtivo(){
        return this.ativo;
    }

    public boolean verificaLogin(ClUsuario usuario){
        DAOUsuario dAOUsuario = new DAOUsuario();
        return dAOUsuario.verificaLogin(usuario);
    }
    public void cadastrarUsuario(ClUsuario usuario) {
        DAOUsuario dAOUsuario = new DAOUsuario();
        dAOUsuario.cadastrarUsuario(usuario);
    }
    public ClUsuario obterUsuarioDetalhes(int codigo) {
        ClUsuario clUsuario = new ClUsuario();
        DAOUsuario dAOUsuario = new DAOUsuario();
        clUsuario = dAOUsuario.obterUsuarioDetalhes(codigo);
        return clUsuario;
    }
    public void editarUsuario(ClUsuario usuario) {
        DAOUsuario dAOUsuario = new DAOUsuario();
        dAOUsuario.editarUsuario(usuario);
    }
    public void removerUsuario(int codigo) {
        DAOUsuario dAOUsuario = new DAOUsuario();
        dAOUsuario.removerUsuario(codigo);
    }
}
```

Pacote com.classes.colecoes

Nome do arquivo: ClBeneficiarioColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOBeneficiario;
import com.classes.ClBeneficiario;

public class ClBeneficiarioColecao {
	public List<ClBeneficiario> buscaBeneficiarios(String filtronome){
		DAOBeneficiario daoBeneficiario	= new DAOBeneficiario();
		List<ClBeneficiario> beneficiarios =
daoBeneficiario.buscaBeneficiarios(filtronome);
		return beneficiarios;
	}
}
```

Nome do arquivo: ClEnderecoColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOEndereco;

public class ClEnderecoColecao {
	public List<Object> buscaListaEnderecos(String CPFCNPJ, String Logradouro){
		DAOEndereco dAOEndereco = new DAOEndereco();
		List<Object> listaEndereco = dAOEndereco.buscaListaEnderecos(CPFCNPJ,
Logradouro);
		return listaEndereco;
	}
}
```

Nome do arquivo: ClGuiaColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOGuia;

public class ClGuiaColecao {
	public List<Object> buscaListaGuiaEmitidas(java.util.Date datainicial, java.util.Date
datafinal, int codigooperadora, String complex){
		DAOGuia dAOGuia = new DAOGuia();
		List<Object> listaguias = dAOGuia.buscaListaGuiaEmitidas(datainicial,
datafinal, codigooperadora, complex);
		return listaguias;
	}
}
```

Nome do arquivo: ClLoteColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOLote;
import com.classes.ClLote;

public class ClLoteColecao {
	public List<ClLote> getLotes(Integer codigoOperadora) {
		DAOLote dAOLote = new DAOLote();
		List<ClLote> lotes = dAOLote.getLotes(Integer.valueOf(codigoOperadora));
		return lotes;
	}
}
```

Nome do arquivo: ClOperadoraColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOOperadora;
import com.classes.ClOperadora;

public class ClOperadoraColecao {
	public List<ClOperadora> buscaOperadoras(String filtronome){
		DAOOperadora daooperadora    = new DAOOperadora();
		List<ClOperadora> operadoras = daooperadora.buscaOperadoras(filtronome);
		return operadoras;
	}
}
```

Nome do arquivo: ClPrestadorColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOPrestador;

public class ClPrestadorColecao {
	public List<Object> buscaPrestadores(String filtroNome, String filtroTipo) {
		DAOPrestador daoPrestador = new DAOPrestador();
		List<Object> prestadores  = daoPrestador.buscaPrestadores(filtroNome,
filtroTipo);
		return prestadores;
	}
}
```

Nome do arquivo: ClUsuarioColecao.java

```
package com.classes.colecoes;

import java.util.List;

import com.DAO.DAOUsuario;
```

```
import com.classes.ClUsuario;

public class ClUsuarioColecao {
        public List<ClUsuario> buscaUsuarios(String filtroNome){
                DAOUsuario daoUsuario   = new DAOUsuario();
                List<ClUsuario> usuarios = daoUsuario.buscaUsuarios(filtroNome);
                return usuarios;
        }
}
```

Pacote com.DAO

Nome do arquivo: DAOAuxiliar.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class DAOAuxiliar {
        private DAOConexao c;

        public DAOAuxiliar() {
                c = DAOConexao.getBancoDeDados();
        }

        public List<String> listaConselhoProfissional() {
                List<String> conselhos = new ArrayList<String>();
                try{
        Connection con = c.getConnection();
        String sql = "SELECT conselho FROM TissConselhoProfissional";
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
         String conselho = rs.getString(1);
         conselhos.add(conselho);
        }
        rs.close();
        stmt.close();
        con.close();
    }
    catch(SQLException e){
        e.printStackTrace();
                }
    catch(Exception e){
        e.printStackTrace();
                }
                return conselhos;
        }

        public List<String> listaTipoLogradouro() {
                List<String> lista = new ArrayList<String>();
                try{
        Connection con = c.getConnection();
```

```
        String sql = "SELECT distinct TissTipLogra_Descricao FROM TissTipoLogradouro";
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
         String conselho = rs.getString(1);
         lista.add(conselho);
        }
        rs.close();
        stmt.close();
        con.close();
      }
      catch(SQLException e){
         e.printStackTrace();
                }
      catch(Exception e){
         e.printStackTrace();
                }
                return lista;
        }
}
```

Nome do arquivo: DAOBeneficiario.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClBeneficiario;

public class DAOBeneficiario {

        private DAOConexao c;

        public DAOBeneficiario(){
                c = DAOConexao.getBancoDeDados();
        }

        public List<ClBeneficiario> buscaBeneficiarios(String filtronome){
                List<ClBeneficiario> beneficiarios = new ArrayList<ClBeneficiario>();
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissBene_nome,TissBene_numcarteira,
TissBene_valcarteira, TissBene_tpplano, TissBene_numcartaonac, TissBene_Codigo FROM
TissBeneficiario WHERE TissBene_nome LIKE ?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, filtronome + "%");
                        ResultSet rs = stmt.executeQuery();
                        while (rs.next()){
                                ClBeneficiario beneficiario = new ClBeneficiario();
                                beneficiario.setNome(rs.getString(1));
                                beneficiario.setNumcarteira(rs.getString(2));
                                beneficiario.setValcarteira(rs.getString(3));
                                beneficiario.setTpplano(rs.getString(4));
```

```
                        beneficiario.setNumcartaonac(rs.getString(5));
                        beneficiario.setCodigo(rs.getInt(6));
                        beneficiarios.add(beneficiario);
                    }
                    rs.close();
                    stmt.close();
                    con.close();
                    }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
                return beneficiarios;
        }

        public void cadastrarBeneficiario(CIBeneficiario beneficiario){
                try{
                        Connection con = c.getConnection();
                        PreparedStatement stmt = con.prepareStatement("INSERT INTO
Tissbeneficiario(TissBene_nome, TissBene_numcarteira, TissBene_valcarteira,
TissBene_tpplano, TissBene_numcartaonac) VALUES (?,?,?,?,?)");
                        stmt.setString(1, beneficiario.getNome());
                        stmt.setString(2,beneficiario.getNumcarteira());
                        stmt.setString(3, beneficiario.getValcarteira());
                        stmt.setString(4, beneficiario.getTpplano());
                        stmt.setString(5, beneficiario.getNumcartaonac());
                        stmt.executeUpdate();
                        stmt.close();
            Statement stmtID = con.createStatement();
            ResultSet rs = stmtID.executeQuery("SELECT LAST_INSERT_ID()");
            rs.next();
            beneficiario.setCodigo(rs.getInt(1));
            rs.close();
            stmtID.close();
                        con.commit();
                        con.close();
                        }
                catch(SQLException e){
                        e.printStackTrace();
                }
        catch(Exception e){
                        e.printStackTrace();
                }
    }

        public void removerBeneficiario(int codigo){
                try{
                        Connection con = c.getConnection();
                        PreparedStatement stmt = con.prepareStatement("DELETE FROM
TissBeneficiario WHERE TissBene_Codigo=?");
                        stmt.setInt(1, codigo);
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                }
                catch(SQLException e) {
            e.printStackTrace();
```

```
                }
        catch(Exception e) {
            e.printStackTrace();
                }
        }

        public ClBeneficiario obterDetalhesBeneficiario(int codigo){
                ClBeneficiario beneficiario = new ClBeneficiario();
                try {
                        Connection con = c.getConnection();
                        String sql = "SELECT TissBene_nome,TissBene_numcarteira,
TissBene_valcarteira, TissBene_tpplano, TissBene_numcartaonac, TissBene_Codigo FROM
TissBeneficiario WHERE TissBene_Codigo=?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setInt(1, codigo);
                        ResultSet rs = stmt.executeQuery();
                        if (rs.next()){
                                beneficiario.setNome(rs.getString(1));
                                beneficiario.setNumcarteira(rs.getString(2));
                                beneficiario.setValcarteira(rs.getString(3));
                                beneficiario.setTpplano(rs.getString(4));
                                beneficiario.setNumcartaonac(rs.getString(5));
                                beneficiario.setCodigo(rs.getInt(6));
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                 }
                        catch(SQLException e) {
                    e.printStackTrace();
                        }
                  catch(Exception e) {
                    e.printStackTrace();
                        }
              return beneficiario;
        }

        public void editarBeneficiario(ClBeneficiario beneficiario){
                try{
                        Connection con = c.getConnection();
                        PreparedStatement stmt = con.prepareStatement("UPDATE
TissBeneficiario SET TissBene_nome=?, TissBene_numcarteira=?, TissBene_valcarteira=?,
TissBene_tpplano=?, TissBene_numcartaonac=? WHERE TissBene_Codigo=?");
                        stmt.setString(1, beneficiario.getNome());
                        stmt.setString(2, beneficiario.getNumcarteira());
                        stmt.setString(3, beneficiario.getValcarteira());
                        stmt.setString(4, beneficiario.getTpplano());
                        stmt.setString(5, beneficiario.getNumcartaonac());
                        stmt.setInt(6, beneficiario.getCodigo());
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
```

```
        }
}
```

Nome do arquivo: DAOConexao.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DAOConexao {
        private Connection con = null;
        private static DAOConexao bd;

        public static synchronized DAOConexao getBancoDeDados(){
    if (bd == null)
       bd = new DAOConexao();
    return bd;
        }

        public Connection getConnection(){
    try{
       Class.forName("com.mysql.jdbc.Driver");
       String url = "jdbc:mysql://localhost:3306/tissnet";
       Connection con = DriverManager.getConnection(url, "tissufpr", "tissufpr");
       con.setAutoCommit(false);
       return con;
    }
    catch (SQLException sqle){
    }
    catch(ClassNotFoundException cnfe){
    }
    return con;
  }
}
```

Nome do arquivo: DAOConfig.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import com.classes.ClConfig;

public class DAOConfig {
        private DAOConexao c;

        public DAOConfig(){
                c = DAOConexao.getBancoDeDados();
        }

        public ClConfig buscaConfig(){
                ClConfig clConfig = new ClConfig();
                try{
```

```
            Connection con = c.getConnection();
            String sql = "SELECT TissConf_ComplexType, TissConf_DirGuias FROM TissConfig";
            PreparedStatement stmt = con.prepareStatement(sql);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
             clConfig.setComplexType(rs.getString(1));
             clConfig.setDirGuias(rs.getString(2));
             break;
            }
            rs.close();
            stmt.close();
            con.close();
        }
        catch(SQLException e){
            e.printStackTrace();
                    }
        catch(Exception e){
            e.printStackTrace();
                    }
                    return clConfig;
            }

            public boolean editarConfig(ClConfig config) {
                    boolean alterou = true;
                    try{
                            Connection con = c.getConnection();
                            PreparedStatement stmt = con.prepareStatement("UPDATE
TissConfig SET TissConf_ComplexType = ?, TissConf_DirGuias = ?");
                            stmt.setString(1, config.getComplexType());
                            stmt.setString(2, config.getDirGuias());
                            stmt.executeUpdate();
                            stmt.close();
                            con.commit();
                            con.close();
                    }
                    catch(SQLException e){
                            alterou = false;
                    }
        catch(Exception e){
            alterou = false;
                    }
        return alterou;
            }
}
```

Nome do arquivo: DAOEndereco.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClEndereco;

public class DAOEndereco {
```

```
	private DAOConexao c;

	public DAOEndereco(){
		c = DAOConexao.getBancoDeDados();
	}

	public List<Object> buscaListaEnderecos(String CPFCNPJ, String Logradouro){
		List<Object> enderecos = new ArrayList<Object>();
		try{
			Connection con = c.getConnection();
			String sql = "SELECT TissPreEnd_CPFCNPJ, TissEndereco_Codigo,
TissEndereco_Logradouro, TissEndereco_Numero, " +
					" TissEndereco_Complemento,
TissEndereco_CEP, TissEndereco_Municipio, TissEndereco_UF, TissEndereco_IBGE, " +
					" TissEndereco_Principal,
TissTipLogra_Codigo from TissEndereco WHERE TissPreEnd_CPFCNPJ = ? AND
TissEndereco_Logradouro LIKE ?";

			PreparedStatement stmt = con.prepareStatement(sql);

			stmt.setString(1, CPFCNPJ);
			stmt.setString(2, Logradouro+ "%");
			ResultSet rs = stmt.executeQuery();
			while(rs.next()){
				ClEndereco clEndereco = new ClEndereco();

	clEndereco.setCFPCNPJ(rs.getString("TissPreEnd_CPFCNPJ"));
				clEndereco.setCodigo(rs.getInt("TissEndereco_Codigo"));

	clEndereco.setTipoLogradouro(rs.getString("TissTipLogra_Codigo"));

	clEndereco.setPrincipal(rs.getString("TissEndereco_Principal"));

	clEndereco.setLogradouro(rs.getString("TissEndereco_Logradouro"));
				clEndereco.setNumero(rs.getString("TissEndereco_Numero"));
				clEndereco.setCep(rs.getString("TissEndereco_CEP"));

	clEndereco.setComplemento(rs.getString("TissEndereco_Complemento"));

	clEndereco.setMunicipio(rs.getString("TissEndereco_Municipio"));
				clEndereco.setUf(rs.getString("TissEndereco_UF"));
				clEndereco.setIbge(rs.getString("TissEndereco_IBGE"));
				enderecos.add(clEndereco);
			}
			rs.close();
			stmt.close();
			con.close();
		}
		catch(SQLException e){
	e.printStackTrace();
		}
	catch(Exception e){
	e.printStackTrace();
		}
		return enderecos;
	}

	public void excluirEndereco(String cpfcnpj, int codigo){
		try{
			Connection con = c.getConnection();
```

```
                        PreparedStatement stmt = con.prepareStatement("DELETE FROM
TissEndereco WHERE TissPreEnd_CPFCNPJ = ? AND TissEndereco_Codigo = ?");
                        stmt.setString(1, cpfcnpj);
                        stmt.setInt(2, codigo);
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                }
                catch(SQLException e) {
        e.printStackTrace();
                }
        catch(Exception e) {
        e.printStackTrace();
                }
        }

        public void cadastrarEndereco(ClEndereco clendereco){
                Connection con = c.getConnection();
                try{
                        String sql = "INSERT INTO TissEndereco (TissPreEnd_CPFCNPJ,
TissEndereco_Codigo, TissEndereco_Logradouro, " +
                                        " TissEndereco_Numero,
TissEndereco_Complemento, TissEndereco_CEP, TissEndereco_Municipio,
TissEndereco_UF, " +
                                        " TissEndereco_IBGE, TissEndereco_Principal,
TissTipLogra_Codigo) " +
                                        " VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, clendereco.getCFPCNPJ());

        clendereco.setCodigo(buscaProximoCodigo(clendereco.getCFPCNPJ()));
                        stmt.setInt(2, clendereco.getCodigo());
                        stmt.setString(3, clendereco.getLogradouro());
                        stmt.setString(4, clendereco.getNumero());
                        stmt.setString(5, clendereco.getComplemento());
                        stmt.setString(6, clendereco.getCep());
                        stmt.setString(7, clendereco.getMunicipio());
                        stmt.setString(8, clendereco.getUf());
                        stmt.setString(9, clendereco.getIbge());
                        String principal = clendereco.getPrincipal();
                        if (principal == null || principal.equals(""))
                                clendereco.setPrincipal("SIM");
                                stmt.setString(10, clendereco.getPrincipal());
                        stmt.setString(11, clendereco.getTipoLogradouro());
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                        if (clendereco.getPrincipal().equals("SIM"))
                                dasabilitaPrincipal(clendereco.getCFPCNPJ(),
clendereco.getCodigo());
                }
                catch (SQLException e) {
                        e.printStackTrace();
                }
                catch (Exception e){
                        e.printStackTrace();
                }
        }
```

```
        private int buscaProximoCodigo(String cpfcnpj){
                int codigo = 0;
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissEndereco_Codigo FROM TissEndereco
WHERE TissPreEnd_CPFCNPJ = ? ORDER BY TissEndereco_Codigo DESC";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, cpfcnpj);
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){
                                codigo = rs.getInt("TissEndereco_Codigo");
                                break;
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
      catch(Exception e){
        e.printStackTrace();
                }
                return codigo + 1;
        }

        public ClEndereco buscaEndereco(String cpfcnpj, int codigo){
                ClEndereco clEndereco = new ClEndereco();
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissPreEnd_CPFCNPJ, TissEndereco_Codigo,
TissEndereco_Logradouro, TissEndereco_Numero, " +
                                        " TissEndereco_Complemento,
TissEndereco_CEP, TissEndereco_Municipio, TissEndereco_UF, TissEndereco_IBGE, " +
                                        " TissEndereco_Principal, TissTipLogra_Codigo
FROM TissEndereco WHERE TissPreEnd_CPFCNPJ = ? AND TissEndereco_Codigo = ?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, cpfcnpj);
                        stmt.setInt(2, codigo);
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){


        clEndereco.setCFPCNPJ(rs.getString("TissPreEnd_CPFCNPJ"));
                                clEndereco.setCodigo(rs.getInt("TissEndereco_Codigo"));

                                String ajusta = rs.getString("TissTipLogra_Codigo");
                                while (ajusta.length()< 3){
                                        ajusta = "0" + ajusta;
                                }
                                String String2 = buscaTipoLogDsc(ajusta);
                                clEndereco.setTipoLogradouro(String2);


        clEndereco.setPrincipal(rs.getString("TissEndereco_Principal"));

        clEndereco.setLogradouro(rs.getString("TissEndereco_Logradouro"));
                                clEndereco.setNumero(rs.getString("TissEndereco_Numero"));
                                clEndereco.setCep(rs.getString("TissEndereco_CEP"));
```

```
        clEndereco.setComplemento(rs.getString("TissEndereco_Complemento"));

        clEndereco.setMunicipio(rs.getString("TissEndereco_Municipio"));
                                clEndereco.setUf(rs.getString("TissEndereco_UF"));
                                clEndereco.setIbge(rs.getString("TissEndereco_IBGE"));
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
      catch(Exception e){
        e.printStackTrace();
                }
                return clEndereco;
        }

        public void atualizarEndereco(ClEndereco clendereco){
                Connection con = c.getConnection();
                try{
                        String sql = "UPDATE TissEndereco SET TissEndereco_Logradouro =
?, TissEndereco_Numero = ?, TissEndereco_Complemento = ?," +
                                        " TissEndereco_CEP = ?, TissEndereco_Municipio
= ?, TissEndereco_UF = ?, TissEndereco_IBGE = ?, " +
                                        "  TissEndereco_Principal = ?,
TissTipLogra_Codigo = ? WHERE TissPreEnd_CPFCNPJ = ? AND TissEndereco_Codigo =
?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, clendereco.getLogradouro());
                        stmt.setString(2, clendereco.getNumero());
                        stmt.setString(3, clendereco.getComplemento());
                        stmt.setString(4, clendereco.getCep());
                        stmt.setString(5, clendereco.getMunicipio());
                        stmt.setString(6, clendereco.getUf());
                        stmt.setString(7, clendereco.getIbge());
                        stmt.setString(8, clendereco.getPrincipal());
                        stmt.setString(9, clendereco.getTipoLogradouro());
                        stmt.setString(10, clendereco.getCFPCNPJ());
                        stmt.setInt(11, clendereco.getCodigo());
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                        if (clendereco.getPrincipal().equals("SIM"))
                           dasabilitaPrincipal(clendereco.getCFPCNPJ(),
clendereco.getCodigo());
                }
                catch (SQLException e) {
                        e.printStackTrace();
                }
                catch (Exception e){
                        e.printStackTrace();
                }
        }

        public ClEndereco buscaEnderecoPrincipal(String cpfcnpj){
                ClEndereco clEndereco = new ClEndereco();
```

```
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissPreEnd_CPFCNPJ, TissEndereco_Codigo,
TissEndereco_Logradouro, TissEndereco_Numero, " +
                                        " TissEndereco_Complemento,
TissEndereco_CEP, TissEndereco_Municipio, TissEndereco_UF, TissEndereco_IBGE, " +
                                        " TissEndereco_Principal, TissTipLogra_Codigo
FROM TissEndereco WHERE TissPreEnd_CPFCNPJ = ? AND TissEndereco_Principal = ?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, cpfcnpj);
                        stmt.setString(2, "SIM");
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){


        clEndereco.setCFPCNPJ(rs.getString("TissPreEnd_CPFCNPJ"));
                                clEndereco.setCodigo(rs.getInt("TissEndereco_Codigo"));
                                String ajusta = rs.getString("TissTipLogra_Codigo");
                                while (ajusta.length()< 3){
                                        ajusta = "0" + ajusta;
                                }
                                String String2 = buscaTipoLogDsc(ajusta);
                                clEndereco.setTipoLogradouro(String2);

        clEndereco.setPrincipal(rs.getString("TissEndereco_Principal"));

        clEndereco.setLogradouro(rs.getString("TissEndereco_Logradouro"));
                                clEndereco.setNumero(rs.getString("TissEndereco_Numero"));
                                clEndereco.setCep(rs.getString("TissEndereco_CEP"));

        clEndereco.setComplemento(rs.getString("TissEndereco_Complemento"));

        clEndereco.setMunicipio(rs.getString("TissEndereco_Municipio"));
                                clEndereco.setUf(rs.getString("TissEndereco_UF"));
                                clEndereco.setIbge(rs.getString("TissEndereco_IBGE"));
                                break;
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
      catch(Exception e){
        e.printStackTrace();
                }
                return clEndereco;
        }

        public String buscaTipoLogCod(String dsc){
                String descricao = "";
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissTipLogra_Codigo FROM
TissTipoLogradouro WHERE TissTipLogra_Descricao = ?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, dsc);
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){
```

```
                        descricao = rs.getString("TissTipLogra_Codigo");
                    }
                    rs.close();
                    stmt.close();
                    con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
                return descricao;
        }

        private String buscaTipoLogDsc(String cod){
                String descricao = "";
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissTipLogra_Descricao FROM
TissTipoLogradouro WHERE TissTipLogra_Codigo = ?";
                    PreparedStatement stmt = con.prepareStatement(sql);
                    stmt.setString(1, cod);
                    ResultSet rs = stmt.executeQuery(); .
                    while(rs.next()){
                            descricao = rs.getString("TissTipLogra_Descricao");
                    }
                    rs.close();
                    stmt.close();
                    con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
                return descricao;
        }

        private void dasabilitaPrincipal(String cpfcnpj, int codigo){
                try{
                        Connection con = c.getConnection();
                        String sql = "UPDATE TissEndereco set TissEndereco_Principal = ?
WHERE TissPreEnd_CPFCNPJ = ? AND TissEndereco_Codigo <> ?";
                    PreparedStatement stmt = con.prepareStatement(sql);
                    stmt.setString(1, "NAO");
                    stmt.setString(2, cpfcnpj);
                    stmt.setInt(3, codigo);
                    stmt.executeUpdate();
                    stmt.close();
                    con.commit();
                    con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
```

```
	}
}
```

Nome do arquivo: DAOGuia.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClGuia;

public class DAOGuia {

	private DAOConexao c;

	public DAOGuia(){
		c = DAOConexao.getBancoDeDados();
	}

	public void cadastrarGuiaEmitida(ClGuia clGuiaEmitida){
		Connection con = c.getConnection();
		try{
			String sql = "INSERT INTO TissGuiasEmitidas
(TissGuias_CodigoOperadora, TissGuias_CTGuia, TissGuias_NumeroGuia, " +
"TissGuias_CodigoBeneficiario,
TissGuias_DataEmissao, TissGuias_Hash) VALUES (?, ?, ?, ?, ?, ?)";
			PreparedStatement stmt = con.prepareStatement(sql);
			stmt.setInt(1, clGuiaEmitida.getClOperadora().getCodigo());
			stmt.setString(2, clGuiaEmitida.getComplexType());
			if (clGuiaEmitida.getNumeroGuia() != null &&
clGuiaEmitida.getNumeroGuia() > 0)
				stmt.setLong(3, clGuiaEmitida.getNumeroGuia());
			else
				stmt.setLong(3,
buscaProxNumGuia(clGuiaEmitida.getClOperadora().getCodigo(),
clGuiaEmitida.getClBeneficiario().getCodigo()));
			stmt.setInt(4, clGuiaEmitida.getClBeneficiario().getCodigo());
			stmt.setDate(5, new
java.sql.Date(clGuiaEmitida.getDataEmissao().getTime()));
			stmt.setString(6, clGuiaEmitida.getHash());
			stmt.executeUpdate();
			stmt.close();
			con.commit();
			con.close();
		}
		catch (SQLException e) {
			e.printStackTrace();
		}
		catch (Exception e){
			e.printStackTrace();
		}
	}
```

```
	public List<Object> buscaListaGuiaEmitidas(java.util.Date datainicial, java.util.Date
datafinal, int codigooperadora, String complex){
		List<Object> guiasEmitidas = new ArrayList<Object>();
		try{
			String sqlWhere = null;
			Connection con = c.getConnection();
			String sql = "SELECT TissGuias_CodigoOperadora,
TissGuias_CTGuia, TissGuias_NumeroGuia, TissGuias_CodigoBeneficiario, " +
					" TissGuias_DataEmissao, TissGuias_Hash from
TissGuiasEmitidas ";
			if (datainicial != null)
				sqlWhere = "TissGuias_DataEmissao >= \"" + new
java.sql.Date(datainicial.getTime()) + "\"";
			if (datainicial != null){
				if (sqlWhere != null)
					sqlWhere = sqlWhere + " AND ";
				sqlWhere = sqlWhere + "TissGuias_DataEmissao <= \"" + new
java.sql.Date(datafinal.getTime()) + "\"";
			}

			if (codigooperadora != 0){
				if (sqlWhere != null)
					sqlWhere = sqlWhere + " AND ";
				sqlWhere = sqlWhere + "TissGuias_CodigoOperadora = " +
codigooperadora;
			}

			if (complex != null && !complex.equals("")){
				if (sqlWhere != null)
					sqlWhere = sqlWhere + " AND ";
				sqlWhere = sqlWhere + "TissGuias_CTGuia = \"" + complex +
"\"";
			}

			if (sqlWhere != null)
				sql = sql + " WHERE " + sqlWhere;
			PreparedStatement stmt = con.prepareStatement(sql);
			ResultSet rs = stmt.executeQuery();
			int codigoOperadora    = 0;
			int codigoBeneficiario = 0;
			while(rs.next()){
				ClGuia clGuiaEmitida = new ClGuia();
				codigoOperadora = rs.getInt("TissGuias_CodigoOperadora");

	clGuiaEmitida.setComplexType(rs.getString("TissGuias_CTGuia"));

	clGuiaEmitida.setNumeroGuia(rs.getLong("TissGuias_NumeroGuia"));
				codigoBeneficiario = rs.getInt("TissGuias_CodigoBeneficiario");

	clGuiaEmitida.setDataEmissao(rs.getDate("TissGuias_DataEmissao"));
				clGuiaEmitida.setHash(rs.getString("TissGuias_Hash"));

				DAOOperadora dAOOperadora = new DAOOperadora();
				DAOBeneficiario dAOBeneficiario = new DAOBeneficiario();

	clGuiaEmitida.setClOperadora(dAOOperadora.obterDetalhesOperadora(codigoOperad
ora));

	clGuiaEmitida.setClBeneficiario(dAOBeneficiario.obterDetalhesBeneficiario(codigoBen
eficiario));
```

```
                        guiasEmitidas.add(clGuiaEmitida);
                    }
                    rs.close();
                    stmt.close();
                    con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
                return guiasEmitidas;
        }

        public String buscaCtGuia(int codigooperadora, int codbene, String hash){
                String ctguia = "";
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissGuias_CTGuia FROM TissGuiasEmitidas
WHERE TissGuias_CodigoOperadora = ? AND " +
                                                " TissGuias_CodigoBeneficiario = ? AND
TissGuias_Hash = ?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setInt(1, codigooperadora);
                        stmt.setInt(2, codbene);
                        stmt.setString(3, hash);
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){
                                ctguia = rs.getString("TissGuias_CTGuia");
                                break;
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
                return ctguia;
        }

        private int buscaProxNumGuia(int codigooperadora, int codbene){
                int prox = 0;
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissGuias_NumeroGuia  FROM
TissGuiasEmitidas WHERE TissGuias_CodigoOperadora = ? AND " +
                                                " TissGuias_CodigoBeneficiario = ? order by
TissGuias_NumeroGuia desc";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setInt(1, codigooperadora);
                        stmt.setInt(2, codbene);
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){
                                prox = rs.getInt("TissGuias_NumeroGuia");
                                break;
```

```
                    }
                    rs.close();
                    stmt.close();
                    con.close();
                }
                catch(SQLException e){
            e.printStackTrace();
                }
        catch(Exception e){
            e.printStackTrace();
                }
                return prox + 1;
        }
}
```

Nome do arquivo: DAOLote.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClLote;

public class DAOLote {
        private DAOConexao c;

        public DAOLote() {
                c = DAOConexao.getBancoDeDados();
        }

        public void cadastraLote(ClLote lote) {
                try{
        Connection con = c.getConnection();
        PreparedStatement stmt = con.prepareStatement("INSERT INTO
TissLoteGuias(TissId_Operadora, InicioLote, FimLote, ProximoNumero) VALUES(?, ?, ?, ?)");
        stmt.setInt(1, lote.getCodigoOperadora());
        stmt.setLong(2, lote.getInicioLote());
        stmt.setLong(3, lote.getFimLote());
        stmt.setLong(4, lote.getInicioLote());
        stmt.executeUpdate();
        stmt.close();
        con.commit();
        con.close();
    }
    catch(SQLException e){
        e.printStackTrace();
                }
    catch(Exception e){
        e.printStackTrace();
                }
        }

        public List<ClLote> getLotes(Integer codigoOperadora) {
                List<ClLote> lotes = new ArrayList<ClLote>();
```

```
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissId_Lote, InicioLote, FimLote,
ProximoNumero FROM TissLoteGuias WHERE TissId_Operadora=? ORDER BY TissId_Lote";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setInt(1, codigoOperadora);
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){
                                CILote lote = new CILote();
                                lote.setCodigoOperadora(codigoOperadora);
                                lote.setIdLote(rs.getInt(1));
                                lote.setInicioLote(rs.getLong(2));
                                lote.setFimLote(rs.getLong(3));
                                lote.setProximo(rs.getLong(4));
                                lotes.add(lote);
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
        catch(Exception e){
        e.printStackTrace();
                }
        return lotes;
        }

        public Long getProximoNumeroLote(Integer codigoOper) {
                try{
                        List<CILote> lotes = getLotes(codigoOper);
                        for (CILote lote : lotes) {
                                if (lote.getProximo() < lote.getFimLote())
                                        return lote.getProximo();
                        }
        }
        catch(Exception e){
        e.printStackTrace();
                }
        return null;
        }

        public void incrementaProximoNumero(CILote lote) {
                try{
        Connection con = c.getConnection();
        PreparedStatement stmt = con.prepareStatement("UPDATE TissLoteGuias SET
ProximoNumero=? WHERE TissId_Lote=?");
        stmt.setLong(1, lote.getProximo() + 1);
        stmt.setInt(2, lote.getIdLote());
        stmt.executeUpdate();
        stmt.close();
        con.commit();
        con.close();
        }
        catch(SQLException e){
        e.printStackTrace();
                }
        catch(Exception e){
        e.printStackTrace();
```

```
                }
        }
}
```

Nome do arquivo: DAOOperadora.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClOperadora;


public class DAOOperadora {
        private DAOConexao c;

        public DAOOperadora() {
                c = DAOConexao.getBancoDeDados();
        }
        public List<ClOperadora> buscaOperadoras(String filtronome){
                List<ClOperadora> operadoras= new ArrayList<ClOperadora>();
                try{
                        Connection con = c.getConnection();
                        String sql = "SELECT TissOpe_regANS,TissOpe_nome,
TissOpe_Codigo, TissOpe_Cnpj, TissOpe_IdPrestador FROM TissOperadora WHERE
TissOpe_nome LIKE ?";
                        PreparedStatement stmt = con.prepareStatement(sql);
                        stmt.setString(1, filtronome + "%");
                        ResultSet rs = stmt.executeQuery();
                        while(rs.next()){
                                ClOperadora operadora = new ClOperadora();
                                operadora.setRegistro(rs.getString(1));
                                operadora.setNome(rs.getString(2));
                                operadora.setCodigo(rs.getInt(3));
                                operadora.setCnpj(rs.getString(4));
                                operadora.setIdPrestador(rs.getString(5));
                                operadoras.add(operadora);
                        }
                        rs.close();
                        stmt.close();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
      catch(Exception e){
        e.printStackTrace();
                }
      return operadoras;
        }

        public void cadastrarOperadora(ClOperadora operadora){
                try{
```

```
        Connection con = c.getConnection();
        PreparedStatement stmt = con.prepareStatement("INSERT INTO
TissOperadora(TissOpe_regANS, TissOpe_nome, TissOpe_cnpj, TissOpe_IdPrestador)
VALUES(?, ?, ?, ?)");
        stmt.setString(1, operadora.getRegistro());
        stmt.setString(2, operadora.getNome());
        stmt.setString(3,operadora.getCnpj());
        stmt.setString(4, operadora.getIdPrestador());
        stmt.executeUpdate();
        stmt.close();
        Statement stmtID = con.createStatement();
        ResultSet rs = stmtID.executeQuery("SELECT LAST_INSERT_ID()");
        rs.next();
        operadora.setCodigo(rs.getInt(1));
        rs.close();
        stmtID.close();
        con.commit();
        con.close();
    }
    catch(SQLException e){
        e.printStackTrace();
            }
    catch(Exception e){
        e.printStackTrace();
            }
        }

        public void removerOperadora(int codigo){
            try{
                Connection con = c.getConnection();
                PreparedStatement stmt = con.prepareStatement("DELETE FROM
TissOperadora WHERE TissOpe_Codigo=?");
                stmt.setInt(1, codigo);
                stmt.executeUpdate();
                stmt.close();
                con.commit();
                con.close();
            }
            catch(SQLException e) {
        e.printStackTrace();
            }
    catch(Exception e) {
        e.printStackTrace();
            }
        }

        public ClOperadora obterDetalhesOperadora(int codigo){
            ClOperadora operadora = new ClOperadora();
            try{
                Connection con = c.getConnection();
                String sql = "SELECT TissOpe_regANS, TissOpe_nome,
TissOpe_Codigo, TissOpe_Cnpj, TissOpe_IdPrestador FROM TissOperadora WHERE
TissOpe_Codigo=?";
                PreparedStatement stmt = con.prepareStatement(sql);
                stmt.setInt(1, codigo);

                ResultSet rs = stmt.executeQuery();
                if (rs.next()){
                    operadora.setRegistro(rs.getString(1));
                    operadora.setNome(rs.getString(2));
```

```
                        operadora.setCodigo(rs.getInt(3));
                        operadora.setCnpj(rs.getString(4));
                        operadora.setIdPrestador(rs.getString(5));
                }
                rs.close();
                stmt.close();
                con.close();
        }
        catch(SQLException e) {
    e.printStackTrace();
        }
catch(Exception e) {
    e.printStackTrace();
        }
return operadora;
    }

    public void editarOperadora(ClOperadora operadora){
        try{
                Connection con = c.getConnection();
                PreparedStatement stmt = con.prepareStatement("UPDATE
TissOperadora SET TissOpe_regANS=?, TissOpe_nome=?, TissOpe_Cnpj=?,
TissOpe_IdPrestador=? WHERE TissOpe_Codigo=?");
                stmt.setString(1, operadora.getRegistro());
                stmt.setString(2, operadora.getNome());
                stmt.setString(3, operadora.getCnpj());
                stmt.setString(4, operadora.getIdPrestador());
                stmt.setInt(5, operadora.getCodigo());
                stmt.executeUpdate();
                stmt.close();
                con.commit();
                con.close();
        }
        catch(SQLException e){
    e.printStackTrace();
        }
catch(Exception e){
    e.printStackTrace();
        }
    }

    public void editarCamposObrigatorios(Integer codigoOperadora, List<String> campos){
        try{
                Connection con = c.getConnection();
                PreparedStatement deleteStmt = con.prepareStatement("DELETE
FROM TissCamposOperadora WHERE tissope_codigo = ?");
                deleteStmt.setInt(1, codigoOperadora);
                deleteStmt.executeUpdate();
                PreparedStatement stmt = con.prepareStatement("INSERT INTO
TissCamposOperadora (TISSOPE_CODIGO, TISSCAMPO) VALUES(?, ?)");
                for (String campo : campos) {
                        stmt.setInt(1, codigoOperadora);
                        stmt.setString(2, campo);
                        stmt.executeUpdate();
                }
                stmt.close();
                con.commit();
                con.close();
        }
        catch(SQLException e){
```

```
            e.printStackTrace();
                    }
            catch(Exception e){
                e.printStackTrace();
                    }
            }

            public List<String> listaCampos(Integer codigoOperadora){
                    List<String> campos = new ArrayList<String>();
                    try{
                            Connection con = c.getConnection();
                            String sql = "SELECT TissCampo FROM TissCamposOperadora
WHERE tissope_codigo = ?";
                            PreparedStatement stmt = con.prepareStatement(sql);
                            stmt.setInt(1, codigoOperadora);
                            ResultSet rs = stmt.executeQuery();
                            while(rs.next()){
                                    campos.add(rs.getString(1));
                            }
                            rs.close();
                            stmt.close();
                            con.close();
                    }
                    catch(SQLException e){
                e.printStackTrace();
                    }
            catch(Exception e){
                e.printStackTrace();
                    }
            return campos;
                }
}
```

Nome do arquivo: DAOPrestador.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClPreFisico;
import com.classes.ClEndereco;
import com.classes.ClPreJuridico;

public class DAOPrestador {

        private DAOConexao c;

        public DAOPrestador() {
                c = DAOConexao.getBancoDeDados();
        }

        public void cadastrarPreFisico(ClPreFisico prefisico){
                Connection conn = c.getConnection();
                try{
```

```
                PreparedStatement stmt = conn.prepareStatement("INSERT INTO
tissprestador (TissPre_cpfcnpj, TissPre_nome,TissPre_cnes, TissPre_codoperadora,
TissPre_cbos, TissPre_conselhoprof, TissPre_numconselho, TissPre_ufconselho,
TissPre_tipo) VALUES (?,?,?,?,?,?,?,?,?) ");
                stmt.setString(1, prefisico.getCpf());
                stmt.setString(2, prefisico.getNome());
                stmt.setString(3, prefisico.getCnes());
                stmt.setString(4, prefisico.getCodoperadora());
                stmt.setString(5, prefisico.getCbos());
                stmt.setString(6, prefisico.getConselhoprof());
                stmt.setString(7, prefisico.getNumconselho());
                stmt.setString(8, prefisico.getUfconselho());
                stmt.setString(9, prefisico.getTipo());
                stmt.executeUpdate();
                stmt.close();
        conn.commit();
                conn.close();
                DAOEndereco dAOEndereco = new DAOEndereco();
                dAOEndereco.cadastrarEndereco(prefisico.getEndereco());
            }
            catch (SQLException e) {
                e.printStackTrace();
                try {
                    conn.rollback();
                } catch (SQLException e1) {
                    e1.printStackTrace();
                }
            }
            catch (Exception e){
                e.printStackTrace();
                try {
                    conn.rollback();
                } catch (SQLException e1) {
                    e1.printStackTrace();
                }
            }
        }

        public void cadastrarPreJuridico(ClPreJuridico prejuridico){
            Connection con = c.getConnection();
            try{
                PreparedStatement stmt = con.prepareStatement("INSERT INTO
tissprestador (TissPre_cpfcnpj, TissPre_nome, TissPre_cnes, TissPre_codoperadora,
TissPre_tipo) VALUES (?, ?, ?, ?, ?)");
                stmt.setString(1, prejuridico.getCnpj());
                stmt.setString(2, prejuridico.getNome());
                stmt.setString(3, prejuridico.getCnes());
                stmt.setString(4, prejuridico.getCodoperadora());
                stmt.setString(5, prejuridico.getTipo());
                stmt.executeUpdate();
                stmt.close();
                con.commit();
                con.close();
                DAOEndereco dAOEndereco = new DAOEndereco();
                dAOEndereco.cadastrarEndereco(prejuridico.getEndereco());
            }
            catch (SQLException e) {
                e.printStackTrace();
                try {
                    con.rollback();
```

```
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }
                }
                catch (Exception e){
                        e.printStackTrace();
                        try {
                                con.rollback();
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }
                }
        }

        public void editarPreFisico(ClPreFisico prefisico){
                try{
                        Connection con = c.getConnection();
                        PreparedStatement stmt = con.prepareStatement("UPDATE
tissprestador SET TissPre_nome=?, TissPre_cnes=?, TissPre_conselhoprof=?,
TissPre_numconselho=?, TissPre_ufconselho=?, TissPre_cbos=?, TissPre_codoperadora=?
WHERE TissPre_cpfcnpj=?");
                        stmt.setString(1, prefisico.getNome());
                        stmt.setString(2, prefisico.getCnes());
                        stmt.setString(3, prefisico.getConselhoprof());
                        stmt.setString(4, prefisico.getNumconselho());
                        stmt.setString(5, prefisico.getUfconselho());
                        stmt.setString(6, prefisico.getCbos());
                        stmt.setString(7, prefisico.getCodoperadora());
                        stmt.setString(8, prefisico.getCpf());
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
      catch(Exception e){
        e.printStackTrace();
                }
        }

        public List<Object> buscaPrestadores(String filtroNome, String filtroTipo) {
                List<Object> prestadores = new ArrayList<Object>();
                try{
         Connection con = c.getConnection();
        String sql = "SELECT TissPre_tipo, TissPre_Nome, TissPre_cpfcnpj FROM
TissPrestador WHERE TissPre_nome LIKE ? AND TissPre_tipo LIKE ?";
        PreparedStatement stmt = con.prepareStatement(sql);
        stmt.setString(1, filtroNome + "%");
        stmt.setString(2, filtroTipo);
        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
         String tipo = rs.getString(1);
         if (tipo.equalsIgnoreCase("fisico")) {
                ClPreFisico preFisico = new ClPreFisico();
                preFisico.setTipo(tipo);
                preFisico.setNome(rs.getString(2));
                preFisico.setCpf(rs.getString(3));
                prestadores.add(preFisico);
```

```
        } else {
                CIPreJuridico preJuridico = new CIPreJuridico();
                preJuridico.setTipo(tipo);
                preJuridico.setNome(rs.getString(2));
                preJuridico.setCnpj(rs.getString(3));
                prestadores.add(preJuridico);
          }
        }
        rs.close();
        stmt.close();
        con.close();
      }
      catch(SQLException e){
        e.printStackTrace();
                }
      catch(Exception e){
        e.printStackTrace();
                }
                return prestadores;
        }

        public CIPreFisico obterPreFisicoDetalhes(String cpf) {
                CIPreFisico preFisico = new CIPreFisico();
                try{
        Connection con = c.getConnection();
        String sql = "SELECT TissPre_nome, TissPre_cpfcnpj, TissPre_cnes,
TissPre_codoperadora, TissPre_cbos, TissPre_conselhoprof," +
        " TissPre_numconselho, TissPre_ufconselho, TissPre_tipo FROM TissPrestador
WHERE TissPre_cpfcnpj = ?";

        PreparedStatement stmt = con.prepareStatement(sql);
        stmt.setString(1, cpf);

        ResultSet rs = stmt.executeQuery();
        if (rs.next()) {
         preFisico.setNome(rs.getString("TissPre_nome"));
         preFisico.setCpf(rs.getString("TissPre_cpfcnpj"));
         preFisico.setCnes(rs.getString("TissPre_cnes"));
         preFisico.setCodoperadora(rs.getString("TissPre_codoperadora"));
         preFisico.setCbos(rs.getString("TissPre_cbos"));
         preFisico.setConselhoprof(rs.getString("TissPre_conselhoprof"));
         preFisico.setNumconselho(rs.getString("TissPre_numconselho"));
         preFisico.setUfconselho(rs.getString("TissPre_ufconselho"));
         preFisico.setTipo(rs.getString("TissPre_tipo"));

         DAOEndereco dAOEndereco = new DAOEndereco();
         preFisico.setEndereco(dAOEndereco.buscaEnderecoPrincipal(preFisico.getCpf()));
        }
        rs.close();

        stmt.close();
        con.close();
      }
      catch(SQLException ex) {
        ex.printStackTrace();
                }
      catch(Exception ex) {
        ex.printStackTrace();
                }
      return preFisico;
```

```
        }

        public void editarPreJuridico(ClPreJuridico prejuridico) {
                try{
                        Connection con = c.getConnection();
                        PreparedStatement stmt = con.prepareStatement("UPDATE
tissprestador SET TissPre_nome=?, TissPre_cnes=?, TissPre_codoperadora=? WHERE
TissPre_cpfcnpj=?");
                        stmt.setString(1, prejuridico.getNome());
                        stmt.setString(2, prejuridico.getCnes());
                        stmt.setString(3, prejuridico.getCodoperadora());
                        stmt.setString(4, prejuridico.getCnpj());
                        stmt.executeUpdate();
                        stmt.close();
                        con.commit();
                        con.close();
                }
                catch(SQLException e){
        e.printStackTrace();
                }
        catch(Exception e){
        e.printStackTrace();
                }
        }

        public ClPreJuridico obterPreJuridicoDetalhes(String cnpj) {
                ClPreJuridico prejuridico = new ClPreJuridico();
                ClEndereco endereco = new ClEndereco();
                prejuridico.setEndereco(endereco);
                try{
                        Connection con=c.getConnection();
                        String sql = "SELECT TissPre_nome, TissPre_cpfcnpj, TissPre_cnes,
TissPre_codoperadora, TissPre_tipo FROM TissPrestador WHERE TissPre_cpfcnpj = ?";
                        PreparedStatement stm = con.prepareStatement(sql);
                        stm.setString(1, cnpj);
                        ResultSet rs = stm.executeQuery();
                        if(rs.next()){
                                prejuridico.setNome(rs.getString("TissPre_nome"));
                                prejuridico.setCnes(rs.getString("TissPre_cnes"));
                                prejuridico.setCnpj(rs.getString("TissPre_cpfcnpj"));

        prejuridico.setCodoperadora(rs.getString("TissPre_codoperadora"));
                                prejuridico.setTipo(rs.getString("TissPre_tipo"));

                                DAOEndereco dAOEndereco = new DAOEndereco();

        prejuridico.setEndereco(dAOEndereco.buscaEnderecoPrincipal(prejuridico.getCnpj()));
                        }
                        rs.close();
                        stm.close();
                        con.close();
                }
                catch(SQLException ex) {
        ex.printStackTrace();
                }
        catch(Exception ex) {
        ex.printStackTrace();
                }
                return prejuridico;
        }
```

```
	public void removerPrestador(String cpfcnpj) {
		try{
			Connection con = c.getConnection();
			PreparedStatement stmt = con.prepareStatement("DELETE FROM
TissPrestador WHERE TissPre_cpfcnpj=?");
			stmt.setString(1, cpfcnpj);
			stmt.executeUpdate();
			stmt.close();
			con.commit();
			con.close();
		}
		catch(SQLException e) {
	e.printStackTrace();
		}
	catch(Exception e) {
	e.printStackTrace();
		}
	}
}
```

Nome do arquivo: DAOUsuario.java

```
package com.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import com.classes.ClUsuario;

public class DAOUsuario {
	private DAOConexao c;

	public DAOUsuario(){
		c = DAOConexao.getBancoDeDados();
	}

	public boolean verificaLogin(ClUsuario usuario){
		boolean chave = false;
		try{
	Connection con = c.getConnection();
	String sql = "SELECT TissUsu_Codigo, TissUsu_Ativo FROM TissUsuario WHERE
TissUsu_Login = ? AND TissUsu_Senha = ? AND TissUsu_Ativo = ?";
	PreparedStatement stmt = con.prepareStatement(sql);
	stmt.setString(1, usuario.getUsuario());
	stmt.setString(2, usuario.getSenha());
	stmt.setString(3, "S");
	ResultSet rs = stmt.executeQuery();
	rs.first();
	if (rs.getRow() > 0){
	 chave = true;
	 usuario.setCodigo(rs.getInt("TissUsu_Codigo"));
	}
	rs.close();
```

```
            stmt.close();
            con.commit();
            con.close();
        }
        catch(SQLException e){
                    }
        catch(Exception e){
                    }
                    return chave;
            }

            public List<ClUsuario> buscaUsuarios(String filtroNome) {
                    List<ClUsuario> usuarios = new ArrayList<ClUsuario>();
                    try{
            Connection con = c.getConnection();
            String sql = "SELECT U.TissUsu_Codigo, U.TissUsu_Ativo, U.TissUsu_Usuario,
U.TissUsu_Login FROM TissUsuario U WHERE U.TissUsu_Usuario LIKE ?";
            PreparedStatement stmt = con.prepareStatement(sql);
            stmt.setString(1, filtroNome + "%");
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
             ClUsuario usuario = new ClUsuario();
             usuario.setCodigo(rs.getInt(1));
             usuario.setAtivo(rs.getString(2));
             usuario.setUsuario(rs.getString(3));
             usuario.setLogin(rs.getString(4));
             usuarios.add(usuario);
            }
            rs.close();
            stmt.close();
            con.close();
        }
        catch(SQLException e){
            e.printStackTrace();
                    }
        catch(Exception e){
            e.printStackTrace();
                    }
                    return usuarios;
            }

            public void cadastrarUsuario(ClUsuario usuario) {
                    try{
            Connection con = c.getConnection();
            PreparedStatement stmt = con.prepareStatement("INSERT INTO
TissUsuario(TissUsu_Usuario, TissUsu_Login, TissUsu_Senha, TissUsu_Ativo) VALUES(?, ?,
?, ?)");
            stmt.setString(1, usuario.getUsuario());
            stmt.setString(2, usuario.getLogin());
            stmt.setString(3, usuario.getSenha());
            stmt.setString(4, usuario.getAtivo());
            stmt.executeUpdate();
            stmt.close();

            Statement stmtID = con.createStatement();
            ResultSet rs = stmtID.executeQuery("SELECT LAST_INSERT_ID()");
             rs.next();
             usuario.setCodigo(rs.getInt(1));
             rs.close();
             stmtID.close();
```

```
            con.commit();
            con.close();
        }
        catch(SQLException e){
                    }
        catch(Exception e){
                    }
            }

            public void removerUsuario(int codigo) {
                    try{
            Connection con = c.getConnection();
            PreparedStatement stmt = con.prepareStatement("DELETE FROM TissUsuario
WHERE TissUsu_Codigo=?");
            stmt.setInt(1, codigo);
            stmt.executeUpdate();
            stmt.close();
            con.commit();
            con.close();
        }
        catch(SQLException e) {
            e.printStackTrace();
                    }
        catch(Exception e) {
            e.printStackTrace();
                    }
            }

            public ClUsuario obterUsuarioDetalhes(int codigo) {
            ClUsuario usuario = new ClUsuario();
                    try{
            Connection con = c.getConnection();
            String sql = "SELECT U.TissUsu_Codigo, U.TissUsu_Ativo, U.TissUsu_Usuario,
U.TissUsu_Login FROM TissUsuario U WHERE U.TissUsu_Codigo = ?";
            PreparedStatement stmt = con.prepareStatement(sql);
            stmt.setInt(1, codigo);

            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
             usuario.setCodigo(rs.getInt(1));
             usuario.setAtivo(rs.getString(2));
             usuario.setUsuario(rs.getString(3));
             usuario.setLogin(rs.getString(4));
            }
            rs.close();

            stmt.close();
            con.close();
        }
        catch(SQLException e) {
            e.printStackTrace();
                    }
        catch(Exception e) {
            e.printStackTrace();
                    }
        return usuario;
            }

            public void editarUsuario(ClUsuario usuario) {
                    try{
```

```
        Connection con = c.getConnection();
        PreparedStatement stmt = con.prepareStatement("UPDATE TissUsuario SET
TissUsu_Usuario=?, TissUsu_Login=?, TissUsu_Ativo=? WHERE TissUsu_Codigo=?");
        stmt.setString(1, usuario.getUsuario());
        stmt.setString(2, usuario.getLogin());
        stmt.setString(3, usuario.getAtivo());
        stmt.setInt(4, usuario.getCodigo());
        stmt.executeUpdate();
        stmt.close();
        con.commit();
        con.close();
    }
    catch(SQLException e){
        e.printStackTrace();
            }
    catch(Exception e){
        e.printStackTrace();
            }
        }
}
```

Pacote com.servlets

Nome do arquivo: SvtConfig.java

```
package com.servlets;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClConfig;

public class SvtConfig extends HttpServlet {
        private static final long serialVersionUID = 1L;
        private String edtComplexType = "";
        private String edtDirGuias    = "";
        private String msgErro        = "";

        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                busca();
                sendLoginForm(response, false);
        }

        private void busca(){
                ClConfig clConfig = new ClConfig();
                clConfig   = clConfig.buscaConfig();
                this.edtComplexType = clConfig.getComplexType();
                this.edtDirGuias    = clConfig.getDirGuias();
        }

        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

```
                String edtSelComplexType = request.getParameter("edtSelComplexType");
                this.edtComplexType     = request.getParameter("edtComplexType");
                this.edtDirGuias        = request.getParameter("edtDirGuias");
                this.msgErro            = "";
                if (botaoAcionado("btnSubmit", request)) {
                        ClConfig clConfig   = new ClConfig();
                        clConfig.setComplexType(edtSelComplexType);
                        clConfig.setDirGuias(this.edtDirGuias);
                        File file = new File(edtSelComplexType);
                        if (!file.exists()){
                                this.msgErro = "Arquivo ComplexType não existe.";
                                this.edtComplexType = "";
                                sendLoginForm(response, true);
                        } else if(!clConfig.editarConfig(clConfig)){
                                        this.msgErro = "Não foi possivel alterar as
configurações.";
                                        sendLoginForm(response, true);
                                } else{
                                        this.msgErro = "As configurações foram salvas.";
                                        this.edtComplexType = edtSelComplexType;
                                        sendLoginForm(response, false);
                        }
                } else{
                        sendLoginForm(response, false);
                }
        }

        private void sendLoginForm(HttpServletResponse response, boolean
withErrorMessage) throws ServletException, IOException{
                response.setContentType("text/html");
                PrintWriter printWriter = response.getWriter();
                printWriter.println("<html>");
                printWriter.println("<head>");
                printWriter.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=ISO-8859-1\">");
                printWriter.println("<title>Configurações</title>");
                printWriter.println("<link rel=\"stylesheet\" type=\"text/css\" media=\"all\"
href=\"niceforms-default.css\" />");
                printWriter.println("</head>");
                printWriter.println("<body>");
                printWriter.println("<form name=\"frmConfig\" method=\"post\"
action=\"SvtConfig.ts\" class=\"niceform\">");
                printWriter.println("<h3>Configurações</h3>");
                printWriter.println("<fieldset><br><legend>Configuração</legend>");
                printWriter.println("<div>");
                printWriter.println("<table>");
                printWriter.println("<tr>");
                printWriter.println("<td><label>Selecionar ComplexType</label></td>");
                printWriter.println("<td><input name=\"edtSelComplexType\" type=\"file\"
id=\"edtSelComplexType\" size=\"50\" maxlength=\"200\" </td>");
                printWriter.println("</tr>");
                printWriter.println("<tr>");
                printWriter.println("<td><label>ComplexType</label></td>");
                printWriter.println("<td> " + this.edtComplexType + "</td>");
                printWriter.println("</tr>");
                printWriter.println("<tr>");
                printWriter.println("<td><label>Salvar guias em</label></td>");
                printWriter.println("<td><input name=\"edtDirGuias\" type=\"text\"
id=\"edtDirGuias\" size=\"50\" maxlength=\"200\" value=\"" + this.edtDirGuias + "\"</td>");
                printWriter.println("</tr>");
```

```
                printWriter.println("</table>");
                printWriter.println("</div>");
                printWriter.println("</fieldset>");
                printWriter.println("<BR>");
                printWriter.println("<input type=\"submit\" name=\"btnSubmit\"
value=\"Confirmar\">");
                printWriter.println("<BR>");
                if (withErrorMessage)
                        printWriter.println(this.msgErro);
                else
                        printWriter.println(this.msgErro);
                printWriter.println("</form>");
                printWriter.println("</body>");
                printWriter.println("</html>");
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request){
                 return request.getParameter(botao)!= null;
        }
}
```

Nome do arquivo: SvtLogin.java

```
package com.servlets;

import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

import com.classes.ClUsuario;


public class SvtLogin extends HttpServlet{
        private static final long serialVersionUID = 1L;
        private String edtUsuario = "";
        private String edtSenha   = "";

        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                this.edtUsuario = "";
                this.edtSenha   = "";
                sendLoginForm(response, false);
        }

        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                this.edtUsuario     = request.getParameter("edtUsuario");
                this.edtSenha       = request.getParameter("edtSenha");
                ClUsuario objUsuario = new ClUsuario();
                objUsuario.setUsuario(this.edtUsuario);
                objUsuario.setSenha(this.edtSenha);
                RequestDispatcher view = request.getRequestDispatcher("menuPrincipal.jsp");
                if (objUsuario.verificaLogin(objUsuario))
                        view.forward(request, response);
                else
                        sendLoginForm(response, true);
        }
```

```
        private void sendLoginForm(HttpServletResponse response, boolean
withErrorMessage) throws ServletException, IOException{
                response.setContentType("text/html");
                PrintWriter printWriter = response.getWriter();
                printWriter.println("<html>");
                printWriter.println("<head>");
                printWriter.println("<title>Login :: TissUFPR</title>");
                printWriter.println("<link rel=\"stylesheet\" type=\"text/css\" media=\"all\"
href=\"niceforms-default.css\" />");
                printWriter.println("</head>");
                printWriter.println("<body>");
                printWriter.println("<form name=\"frmLogin\" method=\"post\"
action=\"login.jsp\" class=\"niceform\">");
                printWriter.println("<center><fieldset><br>");
                printWriter.println("<center><h3>Informe seu usuário e senha</h3></center>");
                printWriter.println("<table width=\"180\" border=\"0\" align=\"center\"
cellpadding=\"0\" cellspacing=\"1\">");
                printWriter.println("<tr>");
                printWriter.println("<td width=\"55\"><label>Login</label></td>");
                if (this.edtUsuario == null || this.edtUsuario.equals("")){
                        printWriter.println("<td width=\"122\"><input name=\"edtUsuario\"
type=\"text\" id=\"edtUsuario\"</td>");
                }else{
                        printWriter.println("<td width=\"122\"><input name=\"edtUsuario\"
type=\"text\" id=\"edtUsuario\" value=\"" + this.edtUsuario + "\"></td>");
                }
                printWriter.println("</tr>");
                printWriter.println("<tr>");
                printWriter.println("<td height=\"18\"><label>Senha</label></td>");
                if (this.edtUsuario == null || this.edtUsuario.equals("")){
                        printWriter.println("<td><input name=\"edtSenha\" type=\"password\"
id=\"edtSenha\" ></td>");
                }else{
                        printWriter.println("<td><input name=\"edtSenha\" type=\"password\"
id=\"edtSenha\" value=\"" + this.edtSenha + "\"></td>");
                }
                printWriter.println("</tr>");
                printWriter.println("<tr>");
                printWriter.println("<td colspan=\"2\"><div align=\"center\">");
                printWriter.println("<input type=\"submit\" name=\"btnSubmit\" id=\"btnSubmit\"
value=\"Confirmar\">");
                printWriter.println("</div></td>");
                printWriter.println("</tr>");
                if (withErrorMessage){
                        printWriter.println("<tr>");
                        printWriter.println("<td colspan=\"2\"><div align=\"left\">");
                        printWriter.println("Login e/ou senha inválido(s).");
                        printWriter.println("</div></td>");
                        printWriter.println("</tr>");
                }
                printWriter.println("</table>");
                printWriter.println("</fieldset></center>");
                printWriter.println("</form>");
                printWriter.println("</body>");
                printWriter.println("</html>");
        }
}
```

Pacote com.servlets.beneficiario

Nome do arquivo: Cadastro.java

```
package com.servlets.beneficiario;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClBeneficiario;

public class Cadastro extends HttpServlet {
        private static final long serialVersionUID = 1L;

   public Cadastro() {
      super();
   }

   protected void service (HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                if(botaoAcionado("CadastrarNovoBeneficiario", request)){
                        request.setAttribute("beneficiario", new ClBeneficiario());
                        request.setAttribute("operacao", "cadastrar");
                        request.getRequestDispatcher("cadastroBen.jsp").forward(request,
response);
                }else if(botaoAcionado("SalvarCadastro", request)){
                        ClBeneficiario beneficiario = new ClBeneficiario();
                        beneficiario.setNome(request.getParameter("nome"));
                        beneficiario.setNumcarteira(request.getParameter("numcarteira"));
                        beneficiario.setValcarteira(request.getParameter("valcarteira"));
                        beneficiario.setTpplano(request.getParameter("tpplano"));

        beneficiario.setNumcartaonac(request.getParameter("numcartaonac"));
                        String operacao = request.getParameter("operacao");
                        if (operacao.equals("cadastrar")){
                                beneficiario.cadastrarBeneficiario(beneficiario);
                        }else{

        beneficiario.setCodigo(Integer.parseInt(request.getParameter("codigo")));
                                beneficiario.editarBeneficiario(beneficiario);
                        }
                        List<ClBeneficiario> listaBeneficiarios = new
ArrayList<ClBeneficiario>();
                        listaBeneficiarios.add(beneficiario);
                        request.setAttribute("listaBeneficiarios", listaBeneficiarios);
                        request.getRequestDispatcher("pesquisaBen.jsp").forward(request,
response);
                }else if ("editar".equals(request.getParameter("operacao"))){
                        String idBeneficiario      = request.getParameter("id");
                        ClBeneficiario beneficiario = new ClBeneficiario();
                        beneficiario            =
beneficiario.obterDetalhesBeneficiario(Integer.parseInt(idBeneficiario));
                        request.setAttribute("beneficiario", beneficiario);
                        request.setAttribute("operacao", "editar");
                        request.getRequestDispatcher("cadastroBen.jsp").forward(request,
response);
```

```
                }
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request){
                return request.getParameter(botao)!=null;
        }
}
```

Nome do arquivo: Excluir.java

```
package com.servlets.beneficiario;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClBeneficiario;

public class Excluir extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                ClBeneficiario clBeneficiario = new ClBeneficiario();
                String idBeneficiario        = request.getParameter("id");
                clBeneficiario.removerBeneficiario(Integer.parseInt(idBeneficiario));
                request.setAttribute("alerta", "Beneficiário removido com sucesso!!!");
                request.getRequestDispatcher("pesquisaBen.jsp").forward(request, response);
        }
}
```

Nome do arquivo: Pesquisa.java

```
package com.servlets.beneficiario;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClBeneficiario;
import com.classes.colecoes.ClBeneficiarioColecao;

public class Pesquisa extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                ClBeneficiarioColecao clBeneficiarioColecao = new ClBeneficiarioColecao();
                if (botaoAcionado("pesquisar", request)) {
                        List<ClBeneficiario> beneficiarios =
clBeneficiarioColecao.buscaBeneficiarios(request.getParameter("nome"));
```

```
                request.setAttribute("listaBeneficiarios", beneficiarios);

            }
            request.getRequestDispatcher("pesquisaBen.jsp").forward(request, response);
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request) {
            return request.getParameter(botao) != null;
        }
}
```

Pacote com.servlets.guia

Nome do arquivo: EscolheBeneficiario.java

```
package com.servlets.guia;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClBeneficiario;
import com.classes.colecoes.ClBeneficiarioColecao;
import com.util.guia.Util;

public class EscolheBeneficiario extends HttpServlet {
        private static final long serialVersionUID = 1L;

    public EscolheBeneficiario() {
        super();
    }

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            String guia                 = request.getParameter("guia");
            String complex              = request.getParameter("complex");
            String operadora            = request.getParameter("operadora");
            String nomeoperadora        =
request.getParameter("nomeoperadora");
            String tituloGuia           = Util.separarPalavras(complex.substring(3,
complex.length()));
            ClBeneficiarioColecao clBeneficiarioColecao = new ClBeneficiarioColecao();
            request.setAttribute("nomeoperadora", nomeoperadora);
            request.setAttribute("nome", request.getParameter("nome"));
            if (botaoAcionado("pesquisar", request)) {
                List<ClBeneficiario> beneficiarios =
clBeneficiarioColecao.buscaBeneficiarios(request.getParameter("nome"));
                request.setAttribute("listaBeneficiarios", beneficiarios);
            }
            request.setAttribute("guia", guia);
            request.setAttribute("complex", complex);
            request.setAttribute("operadora", operadora);
            request.setAttribute("tituloGuia", tituloGuia);
            request.getRequestDispatcher("escolheBen.jsp").forward(request, response);
        }
```

```
	private boolean botaoAcionado(String botao,HttpServletRequest request){
		return request.getParameter(botao)!= null;
	}
}
```

Nome do arquivo: EscolheOperadora.java

```
package com.servlets.guia;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClOperadora;
import com.classes.colecoes.ClOperadoraColecao;
import com.util.guia.Util;

public class EscolheOperadora extends HttpServlet{
	private static final long serialVersionUID = 1L;

  public EscolheOperadora(){
    super();
  }

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		String guia		= request.getParameter("guia");
		String complex		= request.getParameter("complex");
		String tituloGuia	= Util.separarPalavras(complex.substring(3,
complex.length()));
		ClOperadoraColecao clOperadoraColecao = new ClOperadoraColecao();
		request.setAttribute("nome", request.getParameter("nome"));
		if (botaoAcionado("pesquisar",request)){
			List<ClOperadora> operadoras =
clOperadoraColecao.buscaOperadoras(request.getParameter("nome"));
			request.setAttribute("listaOperadoras", operadoras);
		}
		request.setAttribute("guia", guia);
		request.setAttribute("complex", complex);
		request.setAttribute("tituloGuia", tituloGuia);
		request.getRequestDispatcher("escolheOpe.jsp").forward(request, response);
	}

	private boolean botaoAcionado(String botao,HttpServletRequest request){
		return request.getParameter(botao)!= null;
	}
}
```

Nome do arquivo: EscolhePrestador.java

```
package com.servlets.guia;

import java.io.IOException;
```

```
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.colecoes.ClPrestadorColecao;
import com.util.guia.Util;

public class EscolhePrestador extends HttpServlet {
        private static final long serialVersionUID = 1L;

   public EscolhePrestador(){
      super();
   }

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                String guia                     = request.getParameter("guia");
                String complex                  = request.getParameter("complex");
                String operadora                = request.getParameter("operadora");
                String beneficiario             = request.getParameter("beneficiario");
                String nomeoperadora            = request.getParameter("nomeoperadora");
                String nomebene                 = request.getParameter("nomebene");
                String tituloGuia               = Util.separarPalavras(complex.substring(3,
complex.length()));
                ClPrestadorColecao clPrestadorColecao = new ClPrestadorColecao();
                request.setAttribute("nomeoperadora", nomeoperadora);
                request.setAttribute("nomebene", nomebene);
                request.setAttribute("nome", request.getParameter("nome"));
                if (botaoAcionado("pesquisar", request)) {
                        String filtroNome       = request.getParameter("nome");
                        List<Object> prestadores =
clPrestadorColecao.buscaPrestadores(filtroNome, "%");
                        request.setAttribute("listaPrestadores", prestadores);

                }
                request.setAttribute("guia", guia);
                request.setAttribute("complex", complex);
                request.setAttribute("operadora", operadora);
                request.setAttribute("beneficiario", beneficiario);
                request.setAttribute("tituloGuia", tituloGuia);
                request.getRequestDispatcher("escolheCon.jsp").forward(request, response);
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request){
                return request.getParameter(botao)!= null;
        }
}
```

Nome do arquivo: ImprimeGuiaXML.java

```
package com.servlets.guia;

import java.io.IOException;
import java.util.Map;

import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClGuia;
import com.classes.ClGuiaConsulta;
import com.classes.ClGuiaHonorarioIndividual;
import com.classes.ClGuiaResumoInternacao;
import com.classes.ClGuiaSPSADT;
import com.classes.ClGuiaSolicInternacao;
import com.classes.ClOperadora;
import com.util.guia.SAX;
import com.util.guia.Util;

public class ImprimeGuiaXML extends HttpServlet {
	private static final long serialVersionUID = 1L;

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		int codoperadora = Integer.parseInt(request.getParameter("codope"));
		int codbene      = Integer.parseInt(request.getParameter("codbene"));
		String hash      = request.getParameter("hash");

		ClGuia clGuia = new ClGuia();
		String ctGuia = clGuia.buscaCtGuia(codoperadora, codbene, hash);

		ClOperadora clOperadora = new ClOperadora();
		clOperadora             = clOperadora.obterDetalhesOperadora(codoperadora);

		String diretorio = Util.getDiretorioGuias();
		diretorio        = diretorio.trim() + "\\" + clOperadora.getNome().trim() + "\\" +
ctGuia.trim() + "\\" + hash.trim() + ".xml";

		Map<String, String[]> mapaDados = SAX.lerArquivo(diretorio);

		if (ctGuia.equals("ct_guiaConsulta")){
			ClGuiaConsulta guiaC =
ClGuiaConsulta.getGuiaDoMapa(mapaDados);
			request.setAttribute("guia", guiaC);
		}else
		if (ctGuia.equals("ct_guiaHonorarioIndividual")){
			ClGuiaHonorarioIndividual guiaHI =
ClGuiaHonorarioIndividual.getGuiaDoMapa(mapaDados);
			request.setAttribute("guia", guiaHI);
		}else
		if (ctGuia.equals("ct_guiaResumoInternacao")){
			ClGuiaResumoInternacao guiaRI =
ClGuiaResumoInternacao.getGuiaDoMapa(mapaDados);
			request.setAttribute("guia", guiaRI);
		}else
		if (ctGuia.equals("ct_guiaSolicInternacao")){
			ClGuiaSolicInternacao guiaSI =
ClGuiaSolicInternacao.getGuiaDoMapa(mapaDados);
			request.setAttribute("guia", guiaSI);
		}else
		if (ctGuia.equals("ct_guiaSP_SADT")){
			ClGuiaSPSADT guiaSADT =
ClGuiaSPSADT.getGuiaDoMapa(mapaDados);
			request.setAttribute("guia", guiaSADT);
		}
```

```
                ctGuia = ctGuia.substring(3, ctGuia.length());

                request.getRequestDispatcher(ctGuia + ".jsp").forward(request, response);
        }
}
```

Nome do arquivo: PrePreenchimentoGuia.java

```
package com.servlets.guia;

import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClBeneficiario;
import com.classes.ClEndereco;
import com.classes.ClLote;
import com.classes.ClOperadora;
import com.classes.ClPreFisico;
import com.classes.ClPreJuridico;
import com.validacao.Validacao;
import com.validacao.ValidacaoPorCampo;

public class PrePreenchimentoGuia extends HttpServlet {
        private static final long serialVersionUID = 1L;

    public PrePreenchimentoGuia() {
        super();
    }

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                Map<String, Validacao> validacao = ValidacaoPorCampo.getMapValidacao();
                String complex       = request.getParameter("complex");
                String codigoOperadora = request.getParameter("operadora");
                String idBeneficiario  = request.getParameter("beneficiario");
                String contratado      = request.getParameter("contratado");
                Map<String, String[]> dados = new HashMap<String, String[]>();
                ClOperadora operadora     = new ClOperadora();
                List<String> campos       =
operadora.listaCampos(Integer.valueOf(codigoOperadora));
                ClLote ccLote = new ClLote();
                Long proximoNumeroGuia =
ccLote.getProximoNumeroLote(Integer.valueOf(codigoOperadora));
                if (proximoNumeroGuia == null) {

        validacao.put("guiaConsulta.identificacaoGuia.numeroGuiaOperadora",
Validacao.OCULTO);

        validacao.put("guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaOperadora",
Validacao.OCULTO);
```

```
validacao.put("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGui
aOperadora", Validacao.OCULTO);

validacao.put("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroG
uiaOperadora", Validacao.OCULTO);

validacao.put("guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaOpera
dora", Validacao.OCULTO);
        } else {
                dados.put("guiaConsulta.identificacaoGuia.numeroGuiaOperadora",
new String[] { proximoNumeroGuia.toString() });
                validacao.put("guiaConsulta.identificacaoGuia.numeroGuiaPrestador",
Validacao.OCULTO);


dados.put("guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaOperadora", new
String[]{proximoNumeroGuia.toString()});

validacao.put("guiaSP_SADT.identificacaoGuiaSADTSP.numeroGuiaPrestador",
Validacao.OCULTO);


dados.put("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGuiaOp
eradora", new String[]{proximoNumeroGuia.toString()});

validacao.put("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.numeroGui
aPrestador", Validacao.OCULTO);


dados.put("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroGuia
Operadora", new String[]{proximoNumeroGuia.toString()});

validacao.put("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.numeroG
uiaPrestador", Validacao.OCULTO);


dados.put("guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaOperador
a", new String[]{proximoNumeroGuia.toString()});

validacao.put("guiaResumoInternacao.identificacaoGuiaInternacao.numeroGuiaPresta
dor", Validacao.OCULTO);
        }


                operadora =
operadora.obterDetalhesOperadora(Integer.parseInt(codigoOperadora));
                dados.put("guiaConsulta.identificacaoGuia.registroANS", new String[] {
operadora.getRegistro() });
                dados.put("guiaSP_SADT.identificacaoGuiaSADTSP.registroANS", new
String[]{operadora.getRegistro()});

dados.put("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.registroANS",
new String[]{operadora.getRegistro()});

dados.put("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.registroANS",
new String[]{operadora.getRegistro()});
                dados.put("guiaResumoInternacao.identificacaoGuiaInternacao.registroANS",
new String[]{operadora.getRegistro()});
```

```
                CIBeneficiario beneficiario = new CIBeneficiario();
                beneficiario =
beneficiario.obterDetalhesBeneficiario(Integer.parseInt(idBeneficiario));
                dados.put("guiaConsulta.beneficiario.numeroCarteira", new
String[]{beneficiario.getNumcarteira()});
                dados.put("guiaConsulta.beneficiario.nomeBeneficiario", new
String[]{beneficiario.getNome()});
                dados.put("guiaConsulta.beneficiario.nomePlano", new
String[]{beneficiario.getTpplano()});
                dados.put("guiaConsulta.beneficiario.validadeCarteira", new
String[]{beneficiario.getValcarteira()});
                dados.put("guiaConsulta.beneficiario.numeroCNS", new
String[]{beneficiario.getNumcartaonac()});

                dados.put("guiaSP_SADT.dadosBeneficiario.numeroCarteira", new
String[]{beneficiario.getNumcarteira()});
                dados.put("guiaSP_SADT.dadosBeneficiario.nomeBeneficiario", new
String[]{beneficiario.getNome()});
                dados.put("guiaSP_SADT.dadosBeneficiario.nomePlano", new
String[]{beneficiario.getTpplano()});
                dados.put("guiaSP_SADT.dadosBeneficiario.validadeCarteira", new
String[]{beneficiario.getValcarteira()});
                dados.put("guiaSP_SADT.dadosBeneficiario.numeroCNS", new
String[]{beneficiario.getNumcartaonac()});

                dados.put("guiaSolicInternacao.dadosBeneficiario.numeroCarteira", new
String[]{beneficiario.getNumcarteira()});
                dados.put("guiaSolicInternacao.dadosBeneficiario.nomeBeneficiario", new
String[]{beneficiario.getNome()});
                dados.put("guiaSolicInternacao.dadosBeneficiario.nomePlano", new
String[]{beneficiario.getTpplano()});
                dados.put("guiaSolicInternacao.dadosBeneficiario.validadeCarteira", new
String[]{beneficiario.getValcarteira()});
                dados.put("guiaSolicInternacao.dadosBeneficiario.numeroCNS", new
String[]{beneficiario.getNumcartaonac()});

                dados.put("guiaHonorarioIndividual.dadosBeneficiario.numeroCarteira", new
String[]{beneficiario.getNumcarteira()});
                dados.put("guiaHonorarioIndividual.dadosBeneficiario.nomeBeneficiario", new
String[]{beneficiario.getNome()});
                dados.put("guiaHonorarioIndividual.dadosBeneficiario.nomePlano", new
String[]{beneficiario.getTpplano()});
                dados.put("guiaHonorarioIndividual.dadosBeneficiario.validadeCarteira", new
String[]{beneficiario.getValcarteira()});
                dados.put("guiaHonorarioIndividual.dadosBeneficiario.numeroCNS", new
String[]{beneficiario.getNumcartaonac()});

                dados.put("guiaResumoInternacao.dadosBeneficiario.numeroCarteira", new
String[]{beneficiario.getNumcarteira()});
                dados.put("guiaResumoInternacao.dadosBeneficiario.nomeBeneficiario", new
String[]{beneficiario.getNome()});
                dados.put("guiaResumoInternacao.dadosBeneficiario.nomePlano", new
String[]{beneficiario.getTpplano()});
                dados.put("guiaResumoInternacao.dadosBeneficiario.validadeCarteira", new
String[]{beneficiario.getValcarteira()});
                dados.put("guiaResumoInternacao.dadosBeneficiario.numeroCNS", new
String[]{beneficiario.getNumcartaonac()});

                if (contratado.length() == 11) {
                        if (operadora.getIdPrestador().equals("0")) {
```

```
validacao.put("guiaConsulta.dadosContratado.identificacao.CNPJ",
Validacao.OCULTO);

validacao.put("guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperad
ora", Validacao.OCULTO);


validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.CNPJ",
Validacao.OCULTO);

validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.codigoPrestad
orNaOperadora", Validacao.OCULTO);


validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.CNPJ",
Validacao.OCULTO);

validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.codigoPre
stadorNaOperadora", Validacao.OCULTO);


validacao.put("guiaHonorarioIndividual.contratado.identificacao.CNPJ",
Validacao.OCULTO);

validacao.put("guiaHonorarioIndividual.contratado.identificacao.codigoPrestadorNaOp
eradora", Validacao.OCULTO);


validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CNPJ",
Validacao.OCULTO);

validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.codigoPre
stadorNaOperadora", Validacao.OCULTO);
                    } else {

validacao.put("guiaConsulta.dadosContratado.identificacao.CPF",
Validacao.OCULTO);

validacao.put("guiaConsulta.dadosContratado.identificacao.CNPJ",
Validacao.OCULTO);


validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.cpf",
Validacao.OCULTO);

validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.CNPJ",
Validacao.OCULTO);


validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.cpf",
Validacao.OCULTO);

validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.CNPJ",
Validacao.OCULTO);


validacao.put("guiaHonorarioIndividual.contratado.identificacao.CPF",
Validacao.OCULTO);
```

```
        validacao.put("guiaHonorarioIndividual.contratado.identificacao.CNPJ",
Validacao.OCULTO);


        validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CNPJ",
Validacao.OCULTO);
                        }

                        ClPreFisico fisico  = new ClPreFisico();
                        fisico          = fisico.obterPreFisicoDetalhes(contratado);
                        ClEndereco endereco = fisico.getEndereco();
                        dados.put("guiaConsulta.dadosContratado.identificacao.CPF", new
String[]{fisico.getCpf()});

        dados.put("guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperadora
", new String[]{fisico.getCodoperadora()});
                        dados.put("guiaConsulta.dadosContratado.nomeContratado", new
String[]{fisico.getNome()});
                        dados.put("guiaConsulta.dadosContratado.numeroCNES", new
String[]{fisico.getCnes()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.tipoLogradouro", new
String[]{endereco.getTipoLogradouro()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.logradouro", new
String[]{endereco.getLogradouro()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.numero", new
String[]{endereco.getNumero()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.complemento", new
String[]{endereco.getComplemento()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.codigoIBGEMunicipio"
, new String[]{endereco.getIbge()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.municipio", new
String[]{endereco.getMunicipio()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.codigoUF", new
String[]{endereco.getUf()});
                        dados.put("guiaConsulta.dadosContratado.enderecoContratado.cep",
new String[]{endereco.getCep()});


        dados.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.cpf", new
String[]{fisico.getCpf()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.codigoPrestadorN
aOperadora", new String[]{fisico.getCodoperadora()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.nomeContratado", new
String[]{fisico.getNome()});
                        dados.put("guiaSP_SADT.dadosSolicitante.contratado.numeroCNES",
new String[]{fisico.getCnes()});
```

```
dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.tipoLograd
ouro", new String[]{endereco.getTipoLogradouro()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.logradouro
", new String[]{endereco.getLogradouro()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.numero",
new String[]{endereco.getNumero()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.compleme
nto", new String[]{endereco.getComplemento()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigoIBG
EMunicipio", new String[]{endereco.getIbge()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.municipio"
, new String[]{endereco.getMunicipio()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigoUF"
, new String[]{endereco.getUf()});

dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.cep", new
String[]{endereco.getCep()});


dados.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.cpf", new
String[]{fisico.getCpf()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.codigoPresta
dorNaOperadora", new String[]{fisico.getCodoperadora()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.nomeContratado", new
String[]{fisico.getNome()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.numeroCNES", new
String[]{fisico.getCnes()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.tipoL
ogradouro", new String[]{endereco.getTipoLogradouro()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.logra
douro", new String[]{endereco.getLogradouro()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.num
ero", new String[]{endereco.getNumero()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.comp
lemento", new String[]{endereco.getComplemento()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.codig
oIBGEMunicipio", new String[]{endereco.getIbge()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.muni
cipio", new String[]{endereco.getMunicipio()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.codig
oUF", new String[]{endereco.getUf()});
```

```
        dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.cep",
new String[]{endereco.getCep()});

                        dados.put("guiaHonorarioIndividual.contratado.identificacao.CPF", new
String[]{fisico.getCpf()});

        dados.put("guiaHonorarioIndividual.contratado.identificacao.codigoPrestadorNaOperad
ora", new String[]{fisico.getCodoperadora()});
                        dados.put("guiaHonorarioIndividual.contratado.nomeContratado", new
String[]{fisico.getNome()});
                        dados.put("guiaHonorarioIndividual.contratado.numeroCNES", new
String[]{fisico.getCnes()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.tipoLogradouro",
new String[]{endereco.getTipoLogradouro()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.logradouro", new
String[]{endereco.getLogradouro()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.numero", new
String[]{endereco.getNumero()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.complemento",
new String[]{endereco.getComplemento()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.codigoIBGEMunici
pio", new String[]{endereco.getIbge()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.municipio", new
String[]{endereco.getMunicipio()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.codigoUF", new
String[]{endereco.getUf()});

        dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.cep", new
String[]{endereco.getCep()});


        dados.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CPF", new
String[]{fisico.getCpf()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.identificacao.codigoPresta
dorNaOperadora", new String[]{fisico.getCodoperadora()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.nomeContratado", new
String[]{fisico.getNome()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.numeroCNES", new
String[]{fisico.getCnes()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.tipoL
ogradouro", new String[]{endereco.getTipoLogradouro()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.logra
douro", new String[]{endereco.getLogradouro()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.nume
ro", new String[]{endereco.getNumero()});
```

```
        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.compl
emento", new String[]{endereco.getComplemento()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.codig
oIBGEMunicipio", new String[]{endereco.getIbge()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.munic
ipio", new String[]{endereco.getMunicipio()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.codig
oUF", new String[]{endereco.getUf()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.cep",
new String[]{endereco.getCep()});

                    } else { // cnpj
                        if (operadora.getIdPrestador().equals("0")) {

        validacao.put("guiaConsulta.dadosContratado.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperad
ora", Validacao.OCULTO);


        validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.cpf",
Validacao.OCULTO);

        validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.codigoPrestad
orNaOperadora", Validacao.OCULTO);


        validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.cpf",
Validacao.OCULTO);

        validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.codigoPre
stadorNaOperadora", Validacao.OCULTO);


        validacao.put("guiaHonorarioIndividual.contratado.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaHonorarioIndividual.contratado.identificacao.codigoPrestadorNaOp
eradora", Validacao.OCULTO);


        validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.codigoPre
stadorNaOperadora", Validacao.OCULTO);

                        } else { // codigo na operadora

        validacao.put("guiaConsulta.dadosContratado.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaConsulta.dadosContratado.identificacao.CNPJ",
Validacao.OCULTO);
```

```
        validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.cpf",
Validacao.OCULTO);

        validacao.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.CNPJ",
Validacao.OCULTO);


        validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.cpf",
Validacao.OCULTO);

        validacao.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.CNPJ",
Validacao.OCULTO);


        validacao.put("guiaHonorarioIndividual.contratado.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaHonorarioIndividual.contratado.identificacao.CNPJ",
Validacao.OCULTO);


        validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CPF",
Validacao.OCULTO);

        validacao.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CNPJ",
Validacao.OCULTO);
                        }

                        ClPreJuridico juridico = new ClPreJuridico();
                        juridico            = juridico.obterPreJuridicoDetalhes(contratado);
                        ClEndereco endereco    = new ClEndereco();
                        endereco            =
endereco.buscaEnderecoPrincipal(juridico.getCnpj());
                        dados.put("guiaConsulta.dadosContratado.identificacao.CNPJ", new
String[]{juridico.getCnpj()});

        dados.put("guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperadora
", new String[]{juridico.getCodoperadora()});
                        dados.put("guiaConsulta.dadosContratado.nomeContratado", new
String[]{juridico.getNome()});
                        dados.put("guiaConsulta.dadosContratado.numeroCNES", new
String[]{juridico.getCnes()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.tipoLogradouro", new
String[]{endereco.getTipoLogradouro()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.logradouro", new
String[]{endereco.getLogradouro()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.numero", new
String[]{endereco.getNumero()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.complemento", new
String[]{endereco.getComplemento()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.codigoIBGEMunicipio"
, new String[]{endereco.getIbge()});
```

```
        dados.put("guiaConsulta.dadosContratado.enderecoContratado.municipio ", new
String[]{endereco.getMunicipio()});

        dados.put("guiaConsulta.dadosContratado.enderecoContratado.codigoUF", new
String[]{endereco.getUf()});
                        dados.put("guiaConsulta.dadosContratado.enderecoContratado.cep",
new String[]{endereco.getCep()});


        dados.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.CNPJ", new
String[]{juridico.getCnpj()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.identificacao.codigoPrestadorN
aOperadora", new String[]{juridico.getCodoperadora()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.nomeContratado ", new
String[]{juridico.getNome()});
                        dados.put("guiaSP_SADT.dadosSolicitante.contratado.numeroCNES",
new String[]{juridico.getCnes()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.tipoLograd
ouro", new String[]{endereco.getTipoLogradouro()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.logradouro
", new String[]{endereco.getLogradouro()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.numero",
new String[]{endereco.getNumero()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.compleme
nto", new String[]{endereco.getComplemento()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigoIBG
EMunicipio", new String[]{endereco.getIbge()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.municipio"
, new String[]{endereco.getMunicipio()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.codigoUF"
, new String[]{endereco.getUf()});

        dados.put("guiaSP_SADT.dadosSolicitante.contratado.enderecoContratado.cep", new
String[]{endereco.getCep()});


        dados.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.CNPJ", new
String[]{juridico.getCnpj()});

        dados.put("guiaSolicInternacao.dadosSolicitante.contratado.identificacao.codigoPresta
dorNaOperadora", new String[]{juridico.getCodoperadora()});

        dados.put("guiaSolicInternacao.dadosSolicitante.contratado.nomeContratado", new
String[]{juridico.getNome()});

        dados.put("guiaSolicInternacao.dadosSolicitante.contratado.numeroCNES", new
String[]{juridico.getCnes()});

        dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.tipoL
ogradouro", new String[]{endereco.getTipoLogradouro()});
```

```
dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.logradouro", new String[]{endereco.getLogradouro()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.numero", new String[]{endereco.getNumero()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.complemento", new String[]{endereco.getComplemento()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.codigoIBGEMunicipio", new String[]{endereco.getIbge()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.municipio", new String[]{endereco.getMunicipio()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.codigoUF", new String[]{endereco.getUf()});

dados.put("guiaSolicInternacao.dadosSolicitante.contratado.enderecoContratado.cep", new String[]{endereco.getCep()});

                dados.put("guiaHonorarioIndividual.contratado.identificacao.CNPJ", new String[]{juridico.getCnpj()});.

dados.put("guiaHonorarioIndividual.contratado.identificacao.codigoPrestadorNaOperadora", new String[]{juridico.getCodoperadora()});
                dados.put("guiaHonorarioIndividual.contratado.nomeContratado", new String[]{juridico.getNome()});
                dados.put("guiaHonorarioIndividual.contratado.numeroCNES", new String[]{juridico.getCnes()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.tipoLogradouro", new String[]{endereco.getTipoLogradouro()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.logradouro", new String[]{endereco.getLogradouro()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.numero", new String[]{endereco.getNumero()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.complemento", new String[]{endereco.getComplemento()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.codigoIBGEMunicipio", new String[]{endereco.getIbge()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.municipio", new String[]{endereco.getMunicipio()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.codigoUF", new String[]{endereco.getUf()});

dados.put("guiaHonorarioIndividual.contratado.enderecoContratado.cep", new String[]{endereco.getCep()});


dados.put("guiaResumoInternacao.identificacaoExecutante.identificacao.CNPJ", new String[]{juridico.getCnpj()});
```

```
        dados.put("guiaResumoInternacao.identificacaoExecutante.identificacao.codigoPresta
dorNaOperadora", new String[]{juridico.getCodoperadora()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.nomeContratado", new
String[]{juridico.getNome()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.numeroCNES", new
String[]{juridico.getCnes()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.tipoL
ogradouro", new String[]{endereco.getTipoLogradouro()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.logra
douro", new String[]{endereco.getLogradouro()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.nume
ro", new String[]{endereco.getNumero()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.compl
emento", new String[]{endereco.getComplemento()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.codig
oIBGEMunicipio", new String[]{endereco.getIbge()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.munic
ipio", new String[]{endereco.getMunicipio()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.codig
oUF", new String[]{endereco.getUf()});

        dados.put("guiaResumoInternacao.identificacaoExecutante.enderecoContratado.cep",
new String[]{endereco.getCep()});
                }

                request.setAttribute("dados", dados);
                request.setAttribute("complex", complex);
                request.setAttribute("camposobrigatorios", campos);
                request.setAttribute("validacao", validacao);
                request.setAttribute("operadora", codigoOperadora);
                request.setAttribute("beneficiario", idBeneficiario);
                request.getRequestDispatcher("svtGuia.ts").forward(request, response);
        }
}
```

Nome do arquivo: svtConsultarGuia.java

```
package com.servlets.guia;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import com.classes.colecoes.ClGuiaColecao;


public class svtConsultarGuia extends HttpServlet {
	private static final long serialVersionUID = 1L;

	public svtConsultarGuia() {
	super();
  }

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		ClGuiaColecao clGuiaColecao = new ClGuiaColecao();
		String strcodoperadora = null;
		if (botaoAcionado("pesquisar", request)) {
			strcodoperadora = request.getParameter("codoperadora");
			String ctguia = request.getParameter("ctguia");
			int codoperadora = 0;
			if (!strcodoperadora.equals("") && strcodoperadora != null)
				codoperadora = Integer.parseInt(strcodoperadora);

			String strdatainicial = request.getParameter("datainicial");
			String strdatafinal = request.getParameter("datafinal");

			Date dateinicial = new Date();
			Date datefinal = new Date();
		String texto = strdatainicial;
		String formato = "dd/MM/yyyy";
		try {
			dateinicial = new SimpleDateFormat(formato).parse(texto);
			texto      = strdatafinal;
			datefinal   = new SimpleDateFormat(formato).parse(texto);
			} catch (ParseException e) {
			}

			List<Object> listaguias =
clGuiaColecao.buscaListaGuiaEmitidas(dateinicial, datefinal, codoperadora, ctguia);
			request.setAttribute("listaguias", listaguias);
			request.setAttribute("datainicial", request.getParameter("datainicial"));
			request.setAttribute("datafinal", request.getParameter("datafinal"));
			request.setAttribute("codoperadora",
request.getParameter("codoperadora"));
		}
		request.getRequestDispatcher("consultarGuia.jsp").forward(request,
response);
	}

	private boolean botaoAcionado(String botao, HttpServletRequest request){
		return request.getParameter(botao)!= null;
	}
}
```

Nome do arquivo: svtGuia.java

```
package com.servlets.guia;

import java.io.BufferedWriter;
import java.io.File;
```

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClBeneficiario;
import com.classes.ClGuia;
import com.classes.ClGuiaConsulta;
import com.classes.ClGuiaHonorarioIndividual;
import com.classes.ClGuiaResumoInternacao;
import com.classes.ClGuiaSPSADT;
import com.classes.ClGuiaSolicInternacao;
import com.classes.ClLote;
import com.classes.ClOperadora;
import com.util.guia.AgenteMD5;
import com.util.guia.GeradorXML;
import com.util.guia.Util;
import com.validacao.Validacao;


public class svtGuia extends HttpServlet {
	private static final long serialVersionUID = 1L;
	private PrintWriter printWriter;
	private boolean booleanConfirma = false;
	private List<String> camposNaoPreenchidos = new ArrayList<String>();
	private Map<String, String[]> requestGuia = new HashMap<String, String[]>();
	private int CodOperadora    = 0;
	private int CodBeneficiario = 0;
	private String nomeoperadora = "";

	@SuppressWarnings("unchecked")
	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		String complex              = request.getParameter("complex");
		String noXMLGuia            = complex.substring(3, complex.length());
		this.CodOperadora           =
Integer.parseInt(request.getParameter("operadora"));
		this.CodBeneficiario        =
Integer.parseInt(request.getParameter("beneficiario"));
		ClOperadora clOperadora     = new ClOperadora();
		ClBeneficiario clBeneficiario  = new ClBeneficiario();
		printWriter                 = response.getWriter();
		Map<String, String[]> dados    = (Map<String,
String[]>)request.getAttribute("dados");
		Map<String, Validacao> validacao = (Map<String,
Validacao>)request.getAttribute("validacao");
		ClGuia clGuia = new ClGuia();
		clGuia.setNomeGuia(noXMLGuia);
		clGuia.setComplexType(complex);
```

```
        clGuia.setClOperadora(clOperadora.obterDetalhesOperadora(this.CodOperadora));

        clGuia.setClBeneficiario(clBeneficiario.obterDetalhesBeneficiario(this.CodBeneficiario))
;
                clGuia.setGuiaFormularioHTML(response);
                clGuia.setGuiaFormularioHTML(clGuia, dados, validacao);
                printWriter  = clGuia.getGuiaFormularioHTML();
        }

        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                String complex = request.getParameter("complexType");
                String noXMLGuia = request.getParameter("nomeGuia");
                String JSP = noXMLGuia + ".jsp";
                this.CodOperadora    =
Integer.parseInt(request.getParameter("codoperadora"));
                this.CodBeneficiario =
Integer.parseInt(request.getParameter("codbeneficiario"));

                if (botaoAcionado("Imprimir", request)) {

                        if (complex.equals("ct_guiaConsulta")){
                                ClGuiaConsulta guiaC =
ClGuiaConsulta.getGuiaDoRequest(request);
                                request.setAttribute("guia", guiaC);
                        }else
                        if (complex.equals("ct_guiaHonorarioIndividual")){
                                ClGuiaHonorarioIndividual guiaHI =
ClGuiaHonorarioIndividual.getGuiaDoRequest(request);
                                request.setAttribute("guia", guiaHI);
                        }else
                        if (complex.equals("ct_guiaResumoInternacao")){
                                ClGuiaResumoInternacao guiaRI =
ClGuiaResumoInternacao.getGuiaDoRequest(request);
                                request.setAttribute("guia", guiaRI);
                        }else
                        if (complex.equals("ct_guiaSolicInternacao")){
                                ClGuiaSoliInternacao guiaSI =
ClGuiaSoliInternacao.getGuiaDoRequest(request);
                                request.setAttribute("guia", guiaSI);
                        }else
                        if (complex.equals("ct_guiaSP_SADT")){
                                ClGuiaSPSADT guiaSADT =
ClGuiaSPSADT.getGuiaDoRequest(request);
                                request.setAttribute("guia", guiaSADT);
                        }
                        request.getRequestDispatcher(JSP).forward(request, response);
                } else {
                        printWriter = response.getWriter();
                        ClGuia clGuia = new ClGuia();
                        clGuia.setNomeGuia(noXMLGuia);
                        clGuia.setComplexType(complex);
                        ClOperadora clOperadora      = new ClOperadora();
                        ClBeneficiario clBeneficiario = new ClBeneficiario();

        clGuia.setClOperadora(clOperadora.obterDetalhesOperadora(this.CodOperadora));

        clGuia.setClBeneficiario(clBeneficiario.obterDetalhesBeneficiario(this.CodBeneficiario))
;
```

```
                    this.nomeoperadora = clGuia.getClOperadora().getNome();
                    clGuia.setGuiaFormularioHTML(response);

                    if (booleanConfirma = Confirma(request, response, clGuia) ){
                            clGuia.setGuiaFormularioHTML(clGuia, request);
                            printWriter = clGuia.getGuiaFormularioHTML();

                            printWriter.println("<BR>");
                            printWriter.println("<h2>Guia gerada com sucesso!!!</h2>");
                    }else{
                            clGuia.setGuiaFormularioHTML(clGuia, request);
                            printWriter = clGuia.getGuiaFormularioHTML();

                            printWriter.println("<BR>");
                            printWriter.println("<h3>Os seguintes campos obrigatórios não
foram preenchidos:");
                            printWriter.println("<BR>");
                            for (String campo : camposNaoPreenchidos) {
                                    printWriter.println("<BR>");
                                    printWriter.println(campo);
                            }
                            printWriter.println("</h3>");
                    }
            }
    }

    @SuppressWarnings({ "unchecked", "static-access" })
    private boolean Confirma(HttpServletRequest request, HttpServletResponse response,
ClGuia clGuia) throws ServletException, IOException {
            booleanConfirma          = true;
            ClOperadora clOperadora  = new ClOperadora();
            List<String> campos      = clOperadora.listaCampos(this.CodOperadora);
            Map<String,String[]> params = Collections.synchronizedMap((Map<String,
String[]> ) new HashMap<String, String[]>(request.getParameterMap()));
            String complex           = request.getParameter("complexType");
            String nomeGuia          = request.getParameter("nomeGuia");
            camposNaoPreenchidos     = new ArrayList<String>();
            String parametro         = null;
            String composobrigatorios = null;
            for (String campo : campos) {
                    parametro = request.getParameter(campo);
                    for (Map.Entry<String, String[]> e : params.entrySet()){
                            composobrigatorios = e.getKey();
                            if (composobrigatorios.equals(campo)){
                                    if (parametro == null || parametro.equals("")){
                                            camposNaoPreenchidos.add(campo);
                                            booleanConfirma = false;
                                    }
                                    break;
                            }
                    }
            }
            if (booleanConfirma){
                    this.requestGuia = request.getParameterMap();
                    String guiaXML = "";
                    guiaXML = GeradorXML.gerarXML(nomeGuia, complex,
this.requestGuia);
                    try {
                            String diretorio = Util.getDiretorioGuias().trim();
```

```
                                        Util.criarDiretorio(diretorio);
                                        diretorio = diretorio.trim() + "/" + this.nomeoperadora.trim();
                                        Util.criarDiretorio(diretorio);
                                        diretorio = diretorio.trim() + "/" + complex.trim();
                                        Util.criarDiretorio(diretorio);
                                        File dir = new File(diretorio);
                                        if(dir.exists()){
                                                AgenteMD5 agenteMD5         = new AgenteMD5();
                                                String hash = agenteMD5.gerarHashMD5(guiaXML);

                                                BufferedWriter bufferedWriter = new
BufferedWriter(new FileWriter(diretorio +"/" + hash + ".xml"));
                                        bufferedWriter.write(guiaXML);
                                        bufferedWriter.close();
                                        ClLote clLote = new ClLote();

clGuia.setNumeroGuia(clLote.getProximoNumeroLote(Integer.valueOf(this.CodOperadora)));
                                        Date data = new Date();
                                        clGuia.setDataEmissao(data);
                                        clGuia.setHash(hash);
                                        clGuia.cadastrarGuiaEmitida(clGuia);
                                           } else{
                                .                       booleanConfirma = false;
                                                }
                                        } catch (IOException e) {
                                                e.printStackTrace();
                                        }
                                }
                                return booleanConfirma;
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request){
                 return request.getParameter(botao)!= null;
        }
}
```

Pacote com.servlets.operadora

Nome do arquivo: Cadastro.java

```
package com.servlets.operadora;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClCampoGuia;
import com.classes.ClLote;
import com.classes.ClOperadora;
import com.classes.colecoes.ClLoteColecao;
import com.util.guia.Util;
```

```
public class Cadastro extends HttpServlet {
        private static final long serialVersionUID = 1L;

    public Cadastro() {
        super();
    }

        @SuppressWarnings("unchecked")
        protected void service (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

                if (botaoAcionado("CadastrarNovaOperadora",request)){

                        request.setAttribute("operadora", new ClOperadora());
                        request.setAttribute("operacao", "cadastrar");
                        List<ClCampoGuia> camposGuiaConsulta =
Util.getCamposOpcionaisGuia("guiaConsulta", "ct_guiaConsulta");
                        request.setAttribute("camposGuiaConsulta", camposGuiaConsulta);

                        List<ClCampoGuia> camposGuiaSPSADT =
Util.getCamposOpcionaisGuia("guiaSP_SADT", "ct_guiaSP_SADT");
                        request.setAttribute("camposGuiaSP_SADT", camposGuiaSPSADT);

                        List<ClCampoGuia> camposGuiaSolicInternacao =
Util.getCamposOpcionaisGuia("guiaSolicInternacao", "ct_guiaSolicInternacao");
                        request.setAttribute("camposGuiaSolicInternacao",
camposGuiaSolicInternacao);

                        List<ClCampoGuia> camposGuiaResumoInternacao =
Util.getCamposOpcionaisGuia("guiaResumoInternacao", "ct_guiaResumoInternacao");
                        request.setAttribute("camposGuiaResumoInternacao",
camposGuiaResumoInternacao);

                        List<ClCampoGuia> camposGuiaHonorarioIndividual =
Util.getCamposOpcionaisGuia("guiaHonorarioIndividual", "ct_guiaHonorarioIndividual");
                        request.setAttribute("camposGuiaHonorarioIndividual",
camposGuiaHonorarioIndividual);
                        //redireciona para a página .jsp de cadastro operadora
                        request.getRequestDispatcher("cadastroOpe.jsp").forward(request,
response);

                }else if(botaoAcionado("SalvarCadastro",request)){
                        Enumeration parameterNames = request.getParameterNames();
                        List<String> campos      = new ArrayList<String>();
                        for (; parameterNames.hasMoreElements();) {
                                String nomeParametro = (String)
parameterNames.nextElement();
                                if (nomeParametro.startsWith("obr."))
                                        campos.add(nomeParametro.substring(4));
                        }
                        ClOperadora operadora = new ClOperadora();
                        operadora.setRegistro(request.getParameter("registro"));
                        operadora.setNome(request.getParameter("nome"));
                        operadora.setCnpj(request.getParameter("cnpj"));
                        operadora.setIdPrestador(request.getParameter("idprestador"));
                        String operacao = request.getParameter("operacao");
                        if(operacao.equals("cadastrar")){
                                operadora.cadastrarOperadora(operadora);
                        }else{
```

```
operadora.setCodigo(Integer.parseInt(request.getParameter("codigo")));
                        operadora.editarOperadora(operadora);
                }
                operadora.editarCamposObrigatorios(operadora.getCodigo(),
campos);
                String inicioLote = request.getParameter("iniciolote");
                String fimLote = request.getParameter("fimlote");
                if (inicioLote.length() > 0 && fimLote.length() > 0) {
                        ClLote lote = new ClLote();
                        lote.setInicioLote(Long.valueOf(inicioLote));
                        lote.setFimLote(Long.valueOf(fimLote));
                        lote.setCodigoOperadora(operadora.getCodigo());
                        lote.cadastraLote(lote);
                }
                List<ClOperadora> listaOperadoras = new ArrayList<ClOperadora>();

                listaOperadoras.add(operadora);
                request.setAttribute("listaOperadoras", listaOperadoras);
                request.getRequestDispatcher("pesquisaOpe.jsp").forward(request,
response);

        }else if("editar".equals(request.getParameter("operacao"))){
                String idOperadora    = request.getParameter("id");
                ClOperadora operadora = new ClOperadora();
                operadora             =
operadora.obterDetalhesOperadora(Integer.parseInt(idOperadora));
                request.setAttribute("operadora", operadora);
                request.setAttribute("operacao", "editar");

                List<String> camposSelecionados =
operadora.listaCampos(Integer.valueOf(idOperadora));

                List<ClCampoGuia> camposGuiaConsulta =
Util.getCamposOpcionaisGuia("guiaConsulta", "ct_guiaConsulta");
                setCheckedParaCampos(camposGuiaConsulta,
camposSelecionados);
                request.setAttribute("camposGuiaConsulta", camposGuiaConsulta);

                List<ClCampoGuia> camposGuiaSPSADT =
Util.getCamposOpcionaisGuia("guiaSP_SADT", "ct_guiaSP_SADT");
                setCheckedParaCampos(camposGuiaSPSADT,
camposSelecionados);
                request.setAttribute("camposGuiaSP_SADT", camposGuiaSPSADT);

                List<ClCampoGuia> camposGuiaSolicInternacao =
Util.getCamposOpcionaisGuia("guiaSolicInternacao", "ct_guiaSolicInternacao");
                setCheckedParaCampos(camposGuiaSolicInternacao,
camposSelecionados);
                request.setAttribute("camposGuiaSolicInternacao",
camposGuiaSolicInternacao);

                List<ClCampoGuia> camposGuiaResumoInternacao =
Util.getCamposOpcionaisGuia("guiaResumoInternacao", "ct_guiaResumoInternacao");
                setCheckedParaCampos(camposGuiaResumoInternacao,
camposSelecionados);
                request.setAttribute("camposGuiaResumoInternacao",
camposGuiaResumoInternacao);
```

```
                        List<ClCampoGuia> camposGuiaHonorarioIndividual =
Util.getCamposOpcionaisGuia("guiaHonorarioIndividual", "ct_guiaHonorarioIndividual");
                        setCheckedParaCampos(camposGuiaHonorarioIndividual,
camposSelecionados);
                        request.setAttribute("camposGuiaHonorarioIndividual",
camposGuiaHonorarioIndividual);

                        ClLoteColecao clLoteColecao = new ClLoteColecao();
                        List<ClLote> lotes =
clLoteColecao.getLotes(Integer.valueOf(idOperadora));
                        request.setAttribute("lotes", lotes);

                        request.getRequestDispatcher("cadastroOpe.jsp").forward(request,
response);
                }
        }
        private boolean botaoAcionado(String botao, HttpServletRequest request) {
                return request.getParameter(botao)!= null;
        }

        private void setCheckedParaCampos(List<ClCampoGuia> campos, List<String>
nomesSelecionados) {
                for (ClCampoGuia campo : campos ) {
                        boolean selecionado =
nomesSelecionados.contains(campo.getNomeCompleto());
                        campo.setChecked(selecionado);
                }
        }
}
```

Nome do arquivo: Excluir.java

```
package com.servlets.operadora;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClOperadora;

public class Excluir extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                ClOperadora clOperadora = new ClOperadora();
                String idOperadora      = request.getParameter("id");
                clOperadora.removerOperadora(Integer.parseInt(idOperadora));
                request.setAttribute("alerta", "Operadora removida com sucesso!");
                request.getRequestDispatcher("pesquisaOpe.jsp").forward(request, response);
        }
}
```

Nome do arquivo: Pesquisa.java

```
package com.servlets.operadora;
```

```
import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClOperadora;
import com.classes.colecoes.ClOperadoraColecao;

public class Pesquisa extends HttpServlet {
	private static final long serialVersionUID = 1L;

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		ClOperadoraColecao clOperadoraColecao = new ClOperadoraColecao();
		if (botaoAcionado("pesquisar",request)){
			List<ClOperadora> operadoras =
clOperadoraColecao.buscaOperadoras(request.getParameter("nome"));
			request.setAttribute("listaOperadoras", operadoras);
		}
		request.setAttribute("nome", request.getParameter("nome"));
		request.getRequestDispatcher("pesquisaOpe.jsp").forward(request, response);
	}

	private boolean botaoAcionado(String botao,HttpServletRequest request){
		return request.getParameter(botao)!=null;
	}
}
```

Pacote com.servlets.prestador

Nome do arquivo: Excluir.java

```
package com.servlets.prestador;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClPrestador;

public class Excluir extends HttpServlet {
	private static final long serialVersionUID = 1L;

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		ClPrestador clPrestador = new ClPrestador();
		String idPrestador     = request.getParameter("id");
		clPrestador.removerPrestador(idPrestador);
		request.setAttribute("alerta", "Prestador removido com sucesso.");
		request.getRequestDispatcher("pesquisaPre.jsp").forward(request, response);
	}
}
```

Nome do arquivo: Pesquisa.java

```
package com.servlets.prestador;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.colecoes.ClPrestadorColecao;

public class Pesquisa extends HttpServlet {
        private static final long serialVersionUID = 1L;

   public Pesquisa() {
      super();
   }

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                ClPrestadorColecao clPrestadorColecao = new ClPrestadorColecao();
                if (botaoAcionado("pesquisar", request)) {
                        String filtroNome = request.getParameter("nome");
                        String filtroTipo = request.getParameter("opcao");
                        List<Object> prestadores =
clPrestadorColecao.buscaPrestadores(filtroNome, filtroTipo);
                        request.setAttribute("listaPrestadores", prestadores);
                        request.setAttribute("nome", filtroNome);
                        request.setAttribute("opcao", filtroTipo);
                }
                request.getRequestDispatcher("pesquisaPre.jsp").forward(request, response);
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request){
                return request.getParameter(botao)!= null;
        }
}
```

Nome do arquivo: ServletEnderecoPesq.java

```
package com.servlets.prestador;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.colecoes.ClEnderecoColecao;

public class ServletEnderecoPesq extends HttpServlet {
        private static final long serialVersionUID = 1L;
```

```
  public ServletEnderecoPesq() {
    super();
  }

  protected void service(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        ClEnderecoColecao clEnderecoColecao = new ClEnderecoColecao();
        String cpfcnpj                = request.getParameter("cpfcnpj");
        request.setAttribute("cpfcnpj", cpfcnpj);
        request.setAttribute("log", request.getParameter("log"));
        if (botaoAcionado("pesquisar", request)) {
                List<Object> ListaEndereco =
clEnderecoColecao.buscaListaEnderecos(cpfcnpj, request.getParameter("log"));
                request.setAttribute("listaEnderecos", ListaEndereco);
        }
                request.getRequestDispatcher("PesquisaEnder.jsp").forward(request,
response);
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request){
                return request.getParameter(botao)!= null;
        }
}
```

Nome do arquivo: SvtEndereco.java

```
package com.servlets.prestador;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.DAO.DAOAuxiliar;
import com.classes.ClEndereco;

public class SvtEndereco extends HttpServlet {
        private static final long serialVersionUID = 1L;

  public SvtEndereco() {
    super();
  }

  protected void service (HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String cpfcnpj  = request.getParameter("cpfcnpj");
        int    codigo  = Integer.parseInt(request.getParameter("codigo"));
        String operacao = request.getParameter("operacao");
        DAOAuxiliar daoAuxiliar = new DAOAuxiliar();
        ClEndereco clEndereco   = new ClEndereco();

        List<String> lista = daoAuxiliar.listaTipoLogradouro();
                request.setAttribute("listaTipoLogradouro", lista);

        request.setAttribute("cpfcnpj", cpfcnpj);
```

```
        request.setAttribute("codigo", codigo);

        if (botaoAcionado("cadastrarnovo",request)){
                retornaCadastro(request, response);
        } else if (botaoAcionado("SalvarCadastro",request)){
                        String tipoLogradouroDsc = request.getParameter("tipologradouro");
                        String tipoLogradouroCod =
clEndereco.buscaTipoLogCod(tipoLogradouroDsc);
                        clEndereco.setCFPCNPJ(cpfcnpj);
                        clEndereco.setTipoLogradouro(tipoLogradouroCod);
                        clEndereco.setLogradouro(request.getParameter("logradouro"));
                        clEndereco.setNumero(request.getParameter("numero"));
                        clEndereco.setComplemento(request.getParameter("complemento"));
                        clEndereco.setCep(request.getParameter("cep"));
                        clEndereco.setMunicipio(request.getParameter("municipio"));
                        clEndereco.setUf(request.getParameter("uf"));
                        clEndereco.setIbge(request.getParameter("ibge"));
                        clEndereco.setPrincipal(request.getParameter("principal"));
                        if(codigo == 0){
                                clEndereco.cadastrarEndereco(clEndereco);
                } else {
                        clEndereco.setCodigo(codigo);
                        clEndereco.atualizarEndereco(clEndereco);
                }

        request.getRequestDispatcher("PesquisaEnder.jsp?cpfcnpj="+cpfcnpj).forward(request
, response);
        }else if (operacao.equals("editar")){
                clEndereco = clEndereco.buscaEndereco(cpfcnpj, codigo);
                        request.setAttribute("objendereco", clEndereco);
                retornaCadastro(request, response);
        } else if (operacao.equals("excluir")){
                clEndereco.excluirEndereco(cpfcnpj, codigo);
                request.setAttribute("alerta", "Endereço removido com sucesso.");

        request.getRequestDispatcher("PesquisaEnder.jsp?cpfcnpj="+cpfcnpj).forward(request
, response);
        }
   }

   private void retornaCadastro(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException{
        request.getRequestDispatcher("cadastroEndereco.jsp").forward(request, response);
   }

   private boolean botaoAcionado(String botao, HttpServletRequest request) {
                return request.getParameter(botao)!= null;
        }
}
```

Pacote com.servlets.prestador.fisico

Nome do arquivo: Cadastro.java

```
package com.servlets.prestador.fisico;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
```

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.DAO.DAOAuxiliar;
import com.classes.ClEndereco;
import com.classes.ClPreFisico;

public class Cadastro extends HttpServlet {
        private static final long serialVersionUID = 1L;

    public Cadastro() {
        super();
    }

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                DAOAuxiliar daoAuxiliar = new DAOAuxiliar();
                List<String> conselhos  = daoAuxiliar.listaConselhoProfissional();
                request.setAttribute("listaConselhos", conselhos);

                List<String> lista = daoAuxiliar.listaTipoLogradouro();
                request.setAttribute("listaTipoLogradouro", lista);

                if(botaoAcionado("Continuar", request)){
                        ClPreFisico preFisico = new ClPreFisico();
                        preFisico.setCpf(request.getParameter("cpfcnpj"));
                        request.setAttribute("prefisico", preFisico);
                        request.setAttribute("operacao", "cadastrar");

        request.getRequestDispatcher("cadastroPreFisico.jsp").forward(request, response);
                }else if (botaoAcionado("SalvarCadastro",request)){
                        ClPreFisico prefisico   = new ClPreFisico();
                        ClEndereco endereco     = new ClEndereco();
                        String tipoLogradouroDsc = request.getParameter("tipologradouro");
                        String tipoLogradouroCod =
endereco.buscaTipoLogCod(tipoLogradouroDsc);
                        endereco.setCFPCNPJ(request.getParameter("cpf"));
                        endereco.setTipoLogradouro(tipoLogradouroCod);
                        endereco.setLogradouro(request.getParameter("logradouro"));
                        endereco.setNumero(request.getParameter("numero"));
                        endereco.setComplemento(request.getParameter("complemento"));
                        endereco.setCep(request.getParameter("cep"));
                        endereco.setMunicipio(request.getParameter("municipio"));
                        endereco.setUf(request.getParameter("uf"));
                        endereco.setIbge(request.getParameter("ibge"));
                        prefisico.setNome(request.getParameter("nome"));
                        prefisico.setCpf(request.getParameter("cpf"));
                        prefisico.setCnes(request.getParameter("cnes"));
                        prefisico.setCodoperadora(request.getParameter("codoperadora"));
                        prefisico.setCbos(request.getParameter("cbos"));
                        prefisico.setConselhoprof(request.getParameter("conselhoprof"));
                        prefisico.setNumconselho(request.getParameter("numconselho"));
                        prefisico.setUfconselho(request.getParameter("ufconselho"));
                        prefisico.setTipo(request.getParameter("tipo"));
                        prefisico.setEndereco(endereco);
                        String operacao = request.getParameter("operacao");
                        if (operacao.equals("cadastrar")){
```

```
                        prefisico.cadastrarPreFisico(prefisico);
                    }else{
                        endereco.setCFPCNPJ(request.getParameter("cpf"));
                        prefisico.editarPreFisico(prefisico);
                    }
                    List<Object> listaPrestadores = new ArrayList<Object>();
                    listaPrestadores.add(prefisico);
                    request.setAttribute("listaPrestadores", listaPrestadores);
                    request.getRequestDispatcher("pesquisaPre.jsp").forward(request,
response);

            } else if (botaoAcionado("CadastrarEndereco",request)) {
                    String cpf = request.getParameter("cpf");
                    request.setAttribute("cpfcnpj", cpf);

        request.getRequestDispatcher("ServletEnderecoPesq.ts?cpfcnpj="+cpf).forward(reque
st, response);

            } else if ("editar".equals(request.getParameter("operacao"))) {
                    String cpf = request.getParameter("cpf");
                    ClPreFisico preFisico = new ClPreFisico();
                    preFisico = preFisico.obterPreFisicoDetalhes(cpf);
                    request.setAttribute("prefisico", preFisico);

        request.getRequestDispatcher("cadastroPreFisico.jsp").forward(request, response);
                }
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request) {
                return request.getParameter(botao)!= null;
        }
}
```

Pacote com.servlets.prestador.juridico

Nome do arquivo: Cadastro.java

```
package com.servlets.prestador.juridico;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.DAO.DAOAuxiliar;
import com.classes.ClEndereco;
import com.classes.ClPreJuridico;

public class Cadastro extends HttpServlet {
        private static final long serialVersionUID = 1L;

    public Cadastro() {
        super();
    }
```

```
        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                DAOAuxiliar daoAuxiliar = new DAOAuxiliar();
                List<String> lista     = daoAuxiliar.listaTipoLogradouro();
                request.setAttribute("listaTipoLogradouro", lista);
                if(botaoAcionado("Continuar", request)){
                        ClPreJuridico preJuridico = new ClPreJuridico();
                        preJuridico.setCnpj(request.getParameter("cpfcnpj"));
                        request.setAttribute("prejuridico", preJuridico);
                        request.setAttribute("operacao", "cadastrar");

        request.getRequestDispatcher("cadastroPreJuridico.jsp").forward(request, response);
                }else if (botaoAcionado("SalvarCadastro",request)){
                        ClPreJuridico prejuridico = new ClPreJuridico();
                        ClEndereco endereco      = new ClEndereco();
                        String tipoLogradouroDsc = request.getParameter("tipologradouro");
                        String tipoLogradouroCod =
endereco.buscaTipoLogCod(tipoLogradouroDsc);
                        endereco.setCFPCNPJ(request.getParameter("cnpj"));
                        endereco.setTipoLogradouro(tipoLogradouroCod);
                        endereco.setLogradouro(request.getParameter("logradouro"));
                        endereco.setNumero(request.getParameter("numero"));
                        endereco.setComplemento(request.getParameter("complemento"));
                        endereco.setCep(request.getParameter("cep"));
                        endereco.setMunicipio(request.getParameter("municipio"));
                        endereco.setUf(request.getParameter("uf"));
                        endereco.setIbge(request.getParameter("ibge"));
                        prejuridico.setNome(request.getParameter("nome"));
                        prejuridico.setCnpj(request.getParameter("cnpj"));
                        prejuridico.setCnes(request.getParameter("cnes"));
                        prejuridico.setCodoperadora(request.getParameter("codoperadora"));
                        prejuridico.setTipo(request.getParameter("tipo"));
                        prejuridico.setEndereco(endereco);
                        String operacao = request.getParameter("operacao");
                        if (operacao.equals("cadastrar")){
                                prejuridico.cadastrarPreJuridico(prejuridico);
                        }else{
                                endereco.setCFPCNPJ(request.getParameter("cnpj"));
                                prejuridico.editarPreJuridico(prejuridico);
                        }
                        List<Object> listaPrestadores = new ArrayList<Object>();
                        listaPrestadores.add(prejuridico);
                        request.setAttribute("listaPrestadores", listaPrestadores);
                        request.getRequestDispatcher("pesquisaPre.jsp").forward(request,
response);
                        }else if (botaoAcionado("CadastrarEndereco",request)) {
                                String cnpj = request.getParameter("cnpj");
                                request.setAttribute("cpfcnpj", cnpj);

        request.getRequestDispatcher("ServletEnderecoPesq.ts?cpfcnpj="+cnpj).forward(requ
est, response);
                        }else if("editar".equals(request.getParameter("operacao"))){
                                String cnpj = request.getParameter("cnpj");
                                ClPreJuridico prejuridico = new ClPreJuridico();
                                prejuridico          =
prejuridico.obterPreJuridicoDetalhes(cnpj);
                                request.setAttribute("prejuridico", prejuridico);

        request.getRequestDispatcher("cadastroPreJuridico.jsp").forward(request, response);
                        }
```

```
	}

	private boolean botaoAcionado(String botao, HttpServletRequest request) {
		return request.getParameter(botao)!= null;
	}
}
```

Pacote com.servlets.usuario

Nome do arquivo: Cadastro.java

```
package com.servlets.usuario;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClUsuario;

public class Cadastro extends HttpServlet {
	private static final long serialVersionUID = 1L;

  public Cadastro(){
    super();
  }

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		if (botaoAcionado("cadastrarnovo", request)) {
			request.setAttribute("usuario", new ClUsuario());
			request.getRequestDispatcher("cadastroUsu.jsp").forward(request,
response);
		} else if (botaoAcionado("salvar", request)) {
			ClUsuario usuario = new ClUsuario();
			usuario.setUsuario(request.getParameter("nome"));
			usuario.setLogin(request.getParameter("login"));
			usuario.setSenha(request.getParameter("senha"));
			String ativo = request.getParameter("at");
			if ("Ativo".equals(ativo))
				usuario.setAtivo("S");
			else
				usuario.setAtivo("N");
			String codigo = request.getParameter("codigo");
			if (codigo == null || codigo.length() == 0) {
				usuario.cadastrarUsuario(usuario);
			} else {

	usuario.setCodigo(Integer.parseInt(request.getParameter("codigo")));
				usuario.editarUsuario(usuario);
			}
			List<ClUsuario> listaUsuarios = new ArrayList<ClUsuario>();
			listaUsuarios.add(usuario);
			request.setAttribute("listaUsuarios", listaUsuarios);
```

```
                request.getRequestDispatcher("pesquisaUsu.jsp").forward(request,
response);

            } else if ("editar".equals(request.getParameter("operacao"))) {
                String idUsuario  = request.getParameter("id");
                ClUsuario usuario = new ClUsuario();
                usuario = usuario.obterUsuarioDetalhes(Integer.parseInt(idUsuario));
                request.setAttribute("usuario", usuario);
                request.getRequestDispatcher("cadastroUsu.jsp").forward(request,
response);
            }
        }

        private boolean botaoAcionado(String botao, HttpServletRequest request) {
            return request.getParameter(botao) != null;
        }
}
```

Nome do arquivo: Excluir.java

```
package com.servlets.usuario;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClUsuario;

public class Excluir extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            ClUsuario clUsuario = new ClUsuario();
            String idUsuario    = request.getParameter("id");
            clUsuario.removerUsuario(Integer.parseInt(idUsuario));
            request.setAttribute("alerta", "Usuário removido com sucesso.");
            request.getRequestDispatcher("pesquisaUsu.jsp").forward(request, response);
        }
}
```

Nome do arquivo: Pesquisa.java

```
package com.servlets.usuario;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.classes.ClUsuario;
import com.classes.colecoes.ClUsuarioColecao;
```

```
public class Pesquisa extends HttpServlet {
	private static final long serialVersionUID = 1L;

	protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		ClUsuarioColecao clUsuarioColecao = new ClUsuarioColecao();
		if (botaoAcionado("pesquisar", request)) {
			List<ClUsuario> usuarios =
clUsuarioColecao.buscaUsuarios(request.getParameter("nome"));
			request.setAttribute("nome", request.getParameter("nome"));
			request.setAttribute("listaUsuarios", usuarios);
		}
		request.getRequestDispatcher("pesquisaUsu.jsp").forward(request, response);
	}

	private boolean botaoAcionado(String botao, HttpServletRequest request) {
		return request.getParameter(botao) != null;
	}
}
```

Pacote com.util.guia

Nome do arquivo: AgenteMD5.java

```
package com.util.guia;

import java.security.*;

public class AgenteMD5 {

  public AgenteMD5() {
  }
  public static String gerarHashMD5( String pBase ) {
    return gerarHashMD5( pBase, "ISO8859_1" );
  }

   @SuppressWarnings("finally")
      public static String gerarHashMD5( String pBase, String pCharSet ) {
    String wdgs = null;
    try {
      MessageDigest wmd = MessageDigest.getInstance( "MD5" );
      wmd.reset();
      wmd.update( pBase.getBytes( pCharSet ) );
      byte[] wdg  = wmd.digest();
      StringBuffer hexString = new StringBuffer();
      for (int i=0; i < wdg.length; i++ ) {
        String w_dup = Integer.toHexString( 0xFF & wdg[i] );
        if ( w_dup.length() < 2 )
          w_dup = "0" + w_dup;
        hexString.append( w_dup );
      }
      wdgs = hexString.toString();
    } catch (NoSuchAlgorithmException ex) {
      ex.printStackTrace();
    }
    finally {
      return wdgs;
    }
  }
```

```
}
```

Nome do arquivo: GeradorFormularioHTMLGuias.java

```
package com.util.guia;

import java.io.PrintWriter;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import org.apache.xerces.impl.dv.xs.XSSimpleTypeDecl;
import org.apache.xerces.impl.xs.XSComplexTypeDecl;
import org.apache.xerces.impl.xs.XSElementDecl;
import org.apache.xerces.xs.XSComplexTypeDefinition;
import org.apache.xerces.xs.XSFacet;
import org.apache.xerces.xs.XSImplementation;
import org.apache.xerces.xs.XSLoader;
import org.apache.xerces.xs.XSModel;
import org.apache.xerces.xs.XSModelGroup;
import org.apache.xerces.xs.XSObjectList;
import org.apache.xerces.xs.XSParticle;
import org.apache.xerces.xs.XSSimpleTypeDefinition;
import org.apache.xerces.xs.XSTerm;
import org.apache.xerces.xs.XSTypeDefinition;
import org.apache.xerces.xs.datatypes.ObjectList;
import org.w3c.dom.bootstrap.DOMImplementationRegistry;

import com.classes.ClBeneficiario;
import com.classes.ClGuia;
import com.classes.ClOperadora;
import com.validacao.Validacao;
import com.validacao.ValidacaoPorCampo;

public class GeradorFormularioHTMLGuias {

        private Map<String, String[]> requestGuia = new HashMap<String, String[]>();
        private Map<String, Validacao> validacao = ValidacaoPorCampo.getMapValidacao();
        private int codoperadora    = 0;
        private int codbeneficiario = 0;

        public PrintWriter gerarFormularioHTML(ClGuia clGuia, Map<String, String[]> dados,
Map<String, Validacao> validacao) {
                this.requestGuia = dados;
                this.validacao  = validacao;
                ClOperadora clOperadora = new ClOperadora();
                clOperadora = clGuia.getClOperadora();
                this.codoperadora = clOperadora.getCodigo();

                ClBeneficiario clBeneficiario = new ClBeneficiario();
                clBeneficiario = clGuia.getClBeneficiario();
                this.codbeneficiario = clBeneficiario.getCodigo();

                PrintWriter out = clGuia.getGuiaFormularioHTML();
                gerar(out, clGuia.getNomeGuia(), clGuia.getComplexType());
                return out;
        }
```

```
@SuppressWarnings("unchecked")
public PrintWriter gerarFormularioHTML(ClGuia clGuia, HttpServletRequest request) {
	this.requestGuia = request.getParameterMap();
	ClOperadora clOperadora = new ClOperadora();
	clOperadora = clGuia.getClOperadora();
	this.codoperadora = clOperadora.getCodigo();

	ClBeneficiario clBeneficiario = new ClBeneficiario();
	clBeneficiario = clGuia.getClBeneficiario();
	this.codbeneficiario = clBeneficiario.getCodigo();

	PrintWriter out = clGuia.getGuiaFormularioHTML();
	gerar(out, clGuia.getNomeGuia(), clGuia.getComplexType());
	return out;
}

private void listaCampos(XSTypeDefinition typeDef, PrintWriter out, String caminho) {
	if (typeDef instanceof XSComplexTypeDecl) {
		XSComplexTypeDecl complex = (XSComplexTypeDecl) typeDef;
		XSParticle particle = complex.getParticle();
		XSTerm term = particle.getTerm();
		XSModelGroup modelGroup = (XSModelGroup) term;
		XSObjectList campos = modelGroup.getParticles();
		for (int i=0; i < campos.getLength(); i++) {
			XSParticle campo = (XSParticle) campos.item(i);
			if (campo.getTerm() instanceof XSElementDecl) {
				XSElementDecl campoElemento = (XSElementDecl)
campo.getTerm();
				XSTypeDefinition tipoCampo =
campoElemento.getTypeDefinition();
				String nomeCampo = campoElemento.getName();

				if (tipoCampo instanceof XSSimpleTypeDecl) {
					adicionaCampo((XSSimpleTypeDecl)
tipoCampo, nomeCampo, out, caminho);
				} else {
					out.println("<DIV ID=\"" + caminho + "." +
nomeCampo + "\">");
					out.println("<fieldset><br><legend>");
					out.println(Util.separarPalavras(nomeCampo)
+ "</legend>");

					if (campo.getMaxOccursUnbounded()){
						for (int j=0; j < 4; j++){
							out.println("<input
type=\"hidden\" id=\"hdimg" + caminho + "." + nomeCampo + j + "\" value=\"0\"/>");
							out.println("<TABLE
BORDER=2>");
							out.println("<TR> <TD
STYLE=\"width: 600px\">" + Util.separarPalavras(nomeCampo + j) + "</TD>");
							out.println("    <TD> <input
type=\"image\" id=\"img" + caminho + "." + nomeCampo + j + "\"
src=\"img/actionshowfilters.gif\" onClick=\"blocking('" + caminho + "." + nomeCampo + j + "');
return false\" Align=\"right\"/> </TD>");

							out.println("</TR> ");
							out.println("<TR>");
							out.println("<TH
COLSPAN=2>");
```

```
                                                        out.println("    <DIV ID=\"" +
caminho + "." + nomeCampo + j + "\" STYLE=\"display: none\">");
                                                        if (j==0)

        listaCampos(tipoCampo, out, caminho + "." + nomeCampo);
                                                        else

        listaCampos(tipoCampo, out, caminho + "." + nomeCampo + j);
                                                        out.println("    </DIV>");
                                                        out.println("</TH>");
                                                        out.println("</TR>");
                                                        out.println("</TABLE>");
                                                }
                                        } else if (campo.getMaxOccurs() <= 1){
                                                listaCampos(tipoCampo, out, caminho
+ "." + nomeCampo);
                                        }else{
                                                for (int j=0; j < campo.getMaxOccurs();
j++){

                                                        out.println("<input
type=\"hidden\" id=\"hdimg" + caminho + "." + nomeCampo + j + "\" value=\"0\"/>");
                                                        out.println("<TABLE
BORDER=2>");
                                                        out.println("<TR> <TD
STYLE=\"width: 600px\">" + Util.separarPalavras(nomeCampo + j) + "</TD>");
                                                        out.println("    <TD> <input
type=\"image\" id=\"img" + caminho + "." + nomeCampo + j + "\"
src=\"img/actionshowfilters.gif\" onClick=\"blocking('" + caminho + "." + nomeCampo + j + "');
return false\" Align=\"right\"/> </TD>");
                                                        out.println("</TR> ");
                                                        out.println("<TR>");
                                                        out.println("<TH
COLSPAN=2>");
                                                        out.println("    <DIV ID=\"" +
caminho + "." + nomeCampo + j + "\" STYLE=\"display: none\">");
                                                        if (j==0)

        listaCampos(tipoCampo, out, caminho + "." + nomeCampo);
                                                        else

        listaCampos(tipoCampo, out, caminho + "." + nomeCampo + j);
                                                        out.println("    </DIV>");
                                                        out.println("</TH>");
                                                        out.println("</TR>");
                                                        out.println("</TABLE>");
                                                }
                                        }
                                        out.println("</fieldset>");
                                        out.println("</DIV>");
                                }
                        } else {
                                tratarModelGroup((XSModelGroup) campo.getTerm(),
out, caminho);
                        }
                }
        } else if (typeDef instanceof XSSimpleTypeDecl) {
                adicionaCampo((XSSimpleTypeDecl) typeDef, null, out, caminho);
        }
    }
```

```
private void tratarModelGroup(XSModelGroup grupo, PrintWriter out, String caminho) {
        XSObjectList particles = grupo.getParticles();
        for (int i=0; i < particles.getLength(); i++) {
                XSParticle p2 = (XSParticle) particles.item(i);

                XSElementDecl elementDecl = (XSElementDecl) p2.getTerm();
                String nome = elementDecl.getName();
                XSTypeDefinition tipoCampo = elementDecl.getTypeDefinition();

                if (tipoCampo instanceof XSSimpleTypeDecl) {
                        adicionaCampo((XSSimpleTypeDecl) tipoCampo, nome, out,
caminho);
                } else {
                        out.println("<DIV ID=\"" + caminho + "." + nome + "\">");
                        out.println("<fieldset><br><legend>");
                        out.println(Util.separarPalavras(nome) + "</legend>");

                        if (p2.getMaxOccursUnbounded()){
                                for (int j=0; j < 2; j++){
                                        out.println("<input type=\"hidden\" id=\"hdimg"
+ caminho + "." + nome + j + "\" value=\"0\"/>");
                                        out.println("<TABLE BORDER=2>");
                                        out.println("<TR> <TD STYLE=\"width:
600px\">" + Util.separarPalavras(nome + j) + "</TD>");
                                        out.println("    <TD> <input type=\"image\"
id=\"img" + caminho + "." + nome + j + "\" src=\"img/actionshowfilters.gif\" onClick=\"blocking('"
+ caminho + "." + nome + j + "'); return false\" Align=\"right\"/> </TD>");
                                        out.println("</TR> ");
                                        out.println("<TR>");
                                        out.println("<TH COLSPAN=2>");
                                        out.println("    <DIV ID=\"" + caminho + "." +
nome + j + "\" STYLE=\"display: none\">");
                                        if (j==0)
                                                listaCampos(tipoCampo, out, caminho
+ "." + nome);
                                        else
                                                listaCampos(tipoCampo, out, caminho
+ "." + nome + j);
                                        out.println("    </DIV>");
                                        out.println("</TH>");
                                        out.println("</TR>");
                                        out.println("</TABLE>");
                                }
                        } else if (p2.getMaxOccurs() <= 1){
                                listaCampos(tipoCampo, out, caminho + "." + nome);
                        }else{
                                for (int j=0; j < p2.getMaxOccurs(); j++){
                                        out.println("<input type=\"hidden\" id=\"hdimg"
+ caminho + "." + nome + j + "\" value=\"0\"/>");
                                        out.println("<TABLE BORDER=2>");
                                        out.println("<TR> <TD STYLE=\"width:
600px\">" + Util.separarPalavras(nome + j) + "</TD>");
                                        out.println("    <TD> <input type=\"image\"
id=\"img" + caminho + "." + nome + j + "\" src=\"img/actionshowfilters.gif\" onClick=\"blocking('"
+ caminho + "." + nome + j + "'); return false\" Align=\"right\"/> </TD>");
                                        out.println("</TR> ");
                                        out.println("<TR>");
                                        out.println("<TH COLSPAN=2>");
```

```
                                                out.println("    <DIV ID=\"" + caminho + "." +
nome + j + "\" STYLE=\"display: none\">");
                                                if (j==0)
                                                        listaCampos(tipoCampo, out, caminho
+ "." + nome);
                                                else
                                                        listaCampos(tipoCampo, out, caminho
+ "." + nome + j);
                                                out.println("    </DIV>");
                                                out.println("</TH>");
                                                out.println("</TR>");
                                                out.println("</TABLE>");
                                        }
                                }
                                out.println("</fieldset>");
                                out.println("</DIV>");
                        }
                }
        }

        private void adicionaCampo(XSSimpleTypeDecl simpleType, String nomeCampo,
PrintWriter out, String caminho) {
                String parametro = "";
                if (nomeCampo != null) {
                        caminho = caminho + "." + nomeCampo;
                }
                if (this.validacao.get(caminho) == Validacao.OCULTO) {
                        return;
                }
                out.println("<DIV>");
                if (nomeCampo != null) {
                        out.println("<label>"+Util.separarPalavras(nomeCampo)+"</label>");
                }
                if (requestGuia.get(caminho) != null){
                        parametro = requestGuia.get(caminho)[0];
                }

                ObjectList enumeration = simpleType.getActualEnumeration();
                if (enumeration.getLength() > 0) {
                        out.println("<SELECT NAME=\"" + caminho + "\">");
                        for (int i=0; i < enumeration.getLength(); i++) {
                                if (i==0)
                                        out.println("<OPTION VALUE=\"\"> </OPTION>");
                                if (requestGuia != null &&
parametro.equals(enumeration.item(i).toString())){
                                        out.println("<OPTION VALUE=\"" + enumeration.item(i)
+ "\" selected=\"selected\">" + enumeration.item(i) + "</OPTION>");

                                }else{
                                        out.println("<OPTION VALUE=\"" + enumeration.item(i)
+ "\">" + enumeration.item(i) + "</OPTION>");
                                }

                        }
                        out.println("</SELECT>");
                } else {
                        Integer maxLength = null;
                        @SuppressWarnings("unused")
                        Integer minLength = null;
                        XSObjectList facets = simpleType.getFacets();
```

```
                        for (int i=0; i < facets.getLength(); i++) {
                                XSFacet facet = (XSFacet) facets.item(i);
                                if (facet.getFacetKind() ==
XSSimpleTypeDefinition.FACET_MAXLENGTH) {
                                        maxLength =
Integer.parseInt(facet.getLexicalFacetValue());
                                } else if (facet.getFacetKind() ==
XSSimpleTypeDefinition.FACET_MINLENGTH) {
                                        minLength =
Integer.parseInt(facet.getLexicalFacetValue());
                                }
                        }

                        if (maxLength != null && maxLength >= 80) {
                                if (requestGuia.get(caminho) != null ){
                                        out.println("<TEXTAREA ROWS=3 NAME=\"" +
caminho + "\">" + requestGuia.get(caminho)[0] + "</TEXTAREA>");
                                }else{
                                        out.println("<TEXTAREA ROWS=3 NAME=\"" +
caminho + "\"></TEXTAREA>");
                                }

                        } else {
                                if (requestGuia.get(caminho) != null ){
                                        out.print("<INPUT TYPE=\"TEXT\" NAME=\"" +
caminho + "\" VALUE=\"" + requestGuia.get(caminho)[0] + "\"");
                                }else{
                                        out.print("<INPUT TYPE=\"TEXT\" NAME=\"" +
caminho + "\"");
                                }
                                if (maxLength != null) {
                                        out.print(" SIZE=" + maxLength);
                                        out.print(" MAXLENGTH=" + maxLength);
                                }
                                out.println("/>");
                        }

                        Validacao validacao = this.validacao.get(caminho);
                        if (validacao == Validacao.DATA) {
                                String idCal = caminho.replace('.', '_');
                                out.println("<a href=\"javascript:" + idCal + ".popup();\"><img
src=\"img/cal.gif\" width=\"16\" height=\"16\" border=\"0\" alt=\"Clique para selecionar a
data\"></a>");
                                out.println("<script>");
                                out.println("      var " + idCal + " = new calendar1(");
                                out.println("      document.forms[0].elements['" + caminho +
"']);");
                                out.println("</script>");
                        }
                }
                out.println("</DIV>");
        }

        private void gerar(PrintWriter out, String nomeBase, String nomeElemento) {
                try {
                        out.println("<HTML>");
                        out.println("<HEAD>");
                        out.println("<TITLE>" + nomeBase + "</TITLE>");
```

```
                        out.println("<SCRIPT LANGUAGE=\"JavaScript\"
src=\"js/funcoesGuia.js\"> </SCRIPT>");
                        out.println("<SCRIPT LANGUAGE=\"JavaScript\" src=\"calendar1.js\">
</SCRIPT>");
                        out.println("<link rel=\"stylesheet\" type=\"text/css\" media=\"all\"
href=\"niceforms-default.css\" />");
                        out.println("</HEAD>");
                        out.println("<BODY>");

                        out.println("<FORM id=\"formGuiaConsulta\"
name=\"formGuiaConsulta\" method=\"post\" action=\"svtGuia.ts\">");

                        out.println("<input type=\"hidden\" id=\"complexType\"
name=\"complexType\" value=\""+ nomeElemento + "\"/>");
                        out.println("<input type=\"hidden\" id=\"nomeGuia\"
name=\"nomeGuia\" value=\""+ nomeBase + "\"/>");
                        out.println("<input type=\"hidden\" id=\"codoperadora\"
name=\"codoperadora\" value=\""+ this.codoperadora + "\"/>");
                        out.println("<input type=\"hidden\" id=\"codbeneficiario\"
name=\"codbeneficiario\" value=\""+ this.codbeneficiario + "\"/>");
                        out.println("<h3>" + Util.separarPalavras(nomeBase) + "</h3>");

                        // Get DOM Implementation using DOM Registry
                        System.setProperty(DOMImplementationRegistry.PROPERTY,
"org.apache.xerces.dom.DOMXSImplementationSourceImpl");
                        DOMImplementationRegistry registry =
DOMImplementationRegistry.newInstance();

                        XSImplementation impl = (XSImplementation)
registry.getDOMImplementation("XS-Loader");

                        XSLoader schemaLoader = impl.createXSLoader(null);
                        XSModel model = schemaLoader.loadURI(Util.getDiretorioXSD());
                        String caminho = nomeBase;
                        XSTypeDefinition type = model.getTypeDefinition(nomeElemento,
"http://www.ans.gov.br/padroes/tiss/schemas");
                        XSComplexTypeDefinition complex = (XSComplexTypeDefinition)
type;
                        XSParticle particle = complex.getParticle();
                        XSTerm term = particle.getTerm();
                        XSModelGroup modelGroup = (XSModelGroup) term;
                        XSObjectList grupos = modelGroup.getParticles();
                        for (int i=0; i < grupos.getLength(); i++) {

                                XSParticle grupo = (XSParticle) grupos.item(i);
                                XSElementDecl el = (XSElementDecl) grupo.getTerm();
                                String nomeGrupo = el.getName();
                                out.println("<input type=\"hidden\" id=\"hdimg" + nomeGrupo +
"\" value=\"0\"/>");
                                out.println("<TABLE BORDER=2>");
                                out.println("<TR> <TD STYLE=\"width: 600px\">" +
Util.separarPalavras(nomeGrupo) + "</TD>");
                                out.println("    <TD> <input type=\"image\" id=\"img" +
nomeGrupo + "\" src=\"img/actionshowfilters.gif\" onClick=\"blocking('" + nomeGrupo + "');
return false\" Align=\"right\"/> </TD>");

                                out.println("</TR> ");
                                out.println("<TR>");
                                out.println("<TH COLSPAN=2>");
```

```
                                out.println("    <DIV ID=\"" + nomeGrupo + "\" STYLE=\"display:
none\">");
                                XSTypeDefinition typeDef = el.getTypeDefinition();
                                listaCampos(typeDef, out, caminho + "." + nomeGrupo);
                                out.println("    </DIV>");
                                out.println("</TH>");
                                out.println("</TR>");
                                out.println("</TABLE>");
                        }
                        out.println("<BR>");
                        out.println("<input name=\"SalvarCadastro\" type=\"submit\"
value=\"Confirmar\" onClick=\"this.form.target='_self'\">");
                        out.println("<input name=\"Imprimir\" type=\"submit\" value=\"Imprimir\"
onClick=\"this.form.target='_blank'\">");
                        out.println("</FORM>");
                        out.println("</BODY>");
                        out.println("</HTML>");

                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

Nome do arquivo: GeradorXML.java

```
package com.util.guia;

import java.util.Map;

import org.apache.xerces.impl.dv.xs.XSSimpleTypeDecl;
import org.apache.xerces.impl.xs.XSComplexTypeDecl;
import org.apache.xerces.impl.xs.XSElementDecl;
import org.apache.xerces.xs.XSImplementation;
import org.apache.xerces.xs.XSLoader;
import org.apache.xerces.xs.XSModel;
import org.apache.xerces.xs.XSModelGroup;
import org.apache.xerces.xs.XSObjectList;
import org.apache.xerces.xs.XSParticle;
import org.apache.xerces.xs.XSTerm;
import org.apache.xerces.xs.XSTypeDefinition;
import org.w3c.dom.bootstrap.DOMImplementationRegistry;

public class GeradorXML {

        public static String gerarXML(String nomeGuia, String complexType, Map<String,
String[]> valores) {
                StringBuilder xml = new StringBuilder();

                try {
                        // Get DOM Implementation using DOM Registry
                        System.setProperty(DOMImplementationRegistry.PROPERTY,
"org.apache.xerces.dom.DOMXSImplementationSourceImpl");
                        DOMImplementationRegistry registry =
DOMImplementationRegistry.newInstance();

                        XSImplementation impl = (XSImplementation)
registry.getDOMImplementation("XS-Loader");
```

```
                        XSLoader schemaLoader = impl.createXSLoader(null);
                        XSModel model = schemaLoader.loadURI(Util.getDiretorioXSD());
                        xml.append("<" + nomeGuia + ">");
                        XSTypeDefinition type = model.getTypeDefinition(complexType,
"http://www.ans.gov.br/padroes/tiss/schemas");
                        procurarCampos(type, xml, nomeGuia, nomeGuia, valores);
                        xml.append("</" + nomeGuia + ">");

                } catch (Exception e) {
                        e.printStackTrace();
                }

                return xml.toString();
        }

        private static void procurarCampos(XSTypeDefinition typeDef, StringBuilder xml, String
caminho, String nomeGrupo, Map<String, String[]> valores) {
                if (typeDef instanceof XSComplexTypeDecl) {
                        XSComplexTypeDecl complex = (XSComplexTypeDecl) typeDef;
                        XSParticle particle = complex.getParticle();
                        XSTerm term = particle.getTerm();
                        XSModelGroup modelGroup = (XSModelGroup) term;
                        XSObjectList campos = modelGroup.getParticles();
                        for (int i=0; i < campos.getLength(); i++) {
                                XSParticle campo = (XSParticle) campos.item(i);
                                if (campo.getTerm() instanceof XSElementDecl) {

                                        XSElementDecl campoElemento = (XSElementDecl)
campo.getTerm();

                                        XSTypeDefinition tipoCampo =
campoElemento.getTypeDefinition();
                                        String nomeCampo = campoElemento.getName();
                                        xml.append("<" + nomeCampo + ">");

                                        if (tipoCampo instanceof XSSimpleTypeDecl) {
                                                String[] valor = valores.get(caminho + "." +
nomeCampo);
                                                if (valor != null) {
                                                        xml.append(valor[0]);
                                                }
                                        } else {
                                                procurarCampos(tipoCampo, xml, caminho +
"." + nomeCampo, nomeGrupo, valores);
                                        }
                                        xml.append("</" + nomeCampo + ">");
                                } else {
                                        XSModelGroup grupo = (XSModelGroup)
campo.getTerm();
                                        tratarModelGroup(grupo, xml, caminho, valores);
                                }
                        }

                }
        }

        private static void tratarModelGroup(XSModelGroup grupo, StringBuilder xml, String
caminho, Map<String, String[]> valores) {
                XSObjectList particles = grupo.getParticles();
                for (int i=0; i < particles.getLength(); i++) {
```

```
                    XSParticle p2 = (XSParticle) particles.item(i);

                    XSElementDecl elementDecl = (XSElementDecl) p2.getTerm();
                    String nome = elementDecl.getName();
                    XSTypeDefinition tipoCampo = elementDecl.getTypeDefinition();

                    procurarCampos(tipoCampo, xml, caminho, nome, valores);
                }
        }
}
```

Nome do arquivo: SAX.java

```
package com.util.guia;

import java.util.HashMap;
import java.util.Map;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class SAX extends DefaultHandler {

        Map<String, String[]> mapaDados = new HashMap<String, String[]>();
        StringBuffer caminho = new StringBuffer();

        public void startElement(String namespaceURI, String localName,
                        String qName, Attributes atts) {
                if (caminho.length() > 0) {
                        caminho.append(".");
                }

                caminho.append(qName);
        }

        public void characters(char[] ch, int start, int length)
                        throws SAXException {
                String valor = new String(ch, start, length);
                String[] valoresAntigos = mapaDados.get(caminho.toString());
                if (valoresAntigos == null) {
                        String[] valorNovo = new String[] {valor};
                        mapaDados.put(caminho.toString(), valorNovo);
                } else {
                        String[] valorNovo = new String[valoresAntigos.length+1];
                        System.arraycopy(valoresAntigos, 0, valorNovo, 0,
valoresAntigos.length);
                        valorNovo[valorNovo.length-1] = valor;
                }
        }

        public void endElement(String uri, String localName, String name)
                        throws SAXException {
                int pos = caminho.lastIndexOf(".");
                if (pos != -1) {
                        caminho.delete(pos, caminho.length());
```

```
                }
        }

        public static Map<String, String[]> lerArquivo(String nomeArquivo) {
                SAX sax = new SAX();
                try {
                        SAXParser p = SAXParserFactory.newInstance().newSAXParser();
                        p.parse(nomeArquivo, sax);
                } catch (Exception e) {
                        e.printStackTrace();
                }

                return sax.mapaDados;
        }
}
```

Nome do arquivo: Util.java

```
package com.util.guia;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Map;

import org.apache.xerces.impl.dv.xs.XSSimpleTypeDecl;
import org.apache.xerces.impl.xs.XSComplexTypeDecl;
import org.apache.xerces.impl.xs.XSElementDecl;
import org.apache.xerces.xs.XSImplementation;
import org.apache.xerces.xs.XSLoader;
import org.apache.xerces.xs.XSModel;
import org.apache.xerces.xs.XSModelGroup;
import org.apache.xerces.xs.XSObjectList;
import org.apache.xerces.xs.XSParticle;
import org.apache.xerces.xs.XSTerm;
import org.apache.xerces.xs.XSTypeDefinition;
import org.w3c.dom.bootstrap.DOMImplementationRegistry;

import com.DAO.DAOConfig;
import com.classes.ClCampoGuia;
import com.classes.ClConfig;

public class Util {

        public static String separarPalavras(String entrada) {
                StringBuilder resultado = new StringBuilder();
                char primeiro = entrada.charAt(0);
                primeiro = Character.toUpperCase(primeiro);
                resultado.append(primeiro);
                for (int i = 1; i < entrada.length(); i++) {
                        char c = entrada.charAt(i);
                        if (Character.isUpperCase(c) &&
Character.isLowerCase(entrada.charAt(i-1))) {
                                resultado.append(' ');
                        }
                        resultado.append(c);
```

```
                }
                return resultado.toString();
        }

        public static List<ClCampoGuia> getCamposOpcionaisGuia(String nomeBase, String
nomeElemento) {
                List<ClCampoGuia> listaOpcionais = new ArrayList<ClCampoGuia>();
                try {
                        // Get DOM Implementation using DOM Registry
                        System.setProperty(DOMImplementationRegistry.PROPERTY,
"org.apache.xerces.dom.DOMXSImplementationSourceImpl");
                        DOMImplementationRegistry registry =
DOMImplementationRegistry.newInstance();

                        XSImplementation impl = (XSImplementation)
registry.getDOMImplementation("XS-Loader");

                        XSLoader schemaLoader = impl.createXSLoader(null);
                        XSModel model = schemaLoader.loadURI(Util.getDiretorioXSD());
                        XSTypeDefinition type = model.getTypeDefinition(nomeElemento,
"http://www.ans.gov.br/padroes/tiss/schemas");
                        procurarCampos(type, listaOpcionais, nomeBase, nomeBase);

                } catch (Exception e) {
                        e.printStackTrace();
                }
                return listaOpcionais;
        }

        private static void procurarCampos(XSTypeDefinition typeDef, List<ClCampoGuia>
listaOpcionais, String caminho, String nomeGrupo) {
                if (typeDef instanceof XSComplexTypeDecl) {
                        XSComplexTypeDecl complex = (XSComplexTypeDecl) typeDef;
                        XSParticle particle = complex.getParticle();
                        XSTerm term = particle.getTerm();
                        XSModelGroup modelGroup = (XSModelGroup) term;
                        XSObjectList campos = modelGroup.getParticles();
                        for (int i=0; i < campos.getLength(); i++) {
                                XSParticle campo = (XSParticle) campos.item(i);
                                if (campo.getTerm() instanceof XSElementDecl) {

                                        XSElementDecl campoElemento = (XSElementDecl)
campo.getTerm();

                                        XSTypeDefinition tipoCampo =
campoElemento.getTypeDefinition();
                                        String nomeCampo = campoElemento.getName();

                                        if (tipoCampo instanceof XSSimpleTypeDecl) {
                                                ClCampoGuia campoGuia = new
ClCampoGuia();

        campoGuia.setDescricao(Util.separarPalavras(nomeCampo));
                                                campoGuia.setNomeCompleto(caminho + "." +
nomeCampo);
                                                listaOpcionais.add(campoGuia);
                                        } else {
                                                procurarCampos(tipoCampo, listaOpcionais,
caminho + "." + nomeCampo, nomeGrupo);
                                        }
```

```
                                } else {
                                        XSModelGroup grupo = (XSModelGroup)
campo.getTerm();
                                        tratarModelGroup(grupo, listaOpcionais, caminho);
                                }
                        }
                }
        }

        private static void tratarModelGroup(XSModelGroup grupo, List<ClCampoGuia>
listaOpcionais, String caminho) {
                XSObjectList particles = grupo.getParticles();
                for (int i=0; i < particles.getLength(); i++) {
                        XSParticle p2 = (XSParticle) particles.item(i);

                        XSElementDecl elementDecl = (XSElementDecl) p2.getTerm();
                        String nome = elementDecl.getName();
                        XSTypeDefinition tipoCampo = elementDecl.getTypeDefinition();

                        if (tipoCampo instanceof XSSimpleTypeDecl) {
                                ClCampoGuia campoGuia = new ClCampoGuia();
                                campoGuia.setDescricao(Util.separarPalavras(nome));
                                campoGuia.setNomeCompleto(caminho + "." + nome);
                                listaOpcionais.add(campoGuia);
                        } else {
                                procurarCampos(tipoCampo, listaOpcionais, caminho + "." +
nome, nome);
                        }
                }
        }

        public String dataNowString(){
                String dataString = null;
                Date data = new Date();
                SimpleDateFormat formatador = new SimpleDateFormat("dd/MM/yyyy");
                dataString = formatador.format(data).toString();
                return dataString;
        }

        public String HoraMinutoSegundo(){
                String horas = null;
                Calendar calendar = Calendar.getInstance();
                int hora   = calendar.get(Calendar.HOUR_OF_DAY);
                int minuto  = calendar.get(Calendar.MINUTE);
                int segundo = calendar.get(Calendar.SECOND);
                horas = String.valueOf(hora) + ":" + String.valueOf(minuto) + ":" +
String.valueOf(segundo);
                return horas;
        }

        public static String getDiretorioXSD(){
                DAOConfig dAOConfig = new DAOConfig();
                ClConfig clConfig = dAOConfig.buscaConfig();
                return clConfig.getComplexType();
        }

        public static String getDiretorioGuias(){
                DAOConfig dAOConfig = new DAOConfig();
                ClConfig clConfig = dAOConfig.buscaConfig();
                return clConfig.getDirGuias();
```

```
	}

	public static String getPrimeiro(Map<String, String[]> request,   String nome) {
		String[] valor = request.get(nome);
		if (valor == null) {
			return null;
		}
		return valor[0];
	}

	public static boolean criarDiretorio(String diretorio){
		boolean b = false;
		File dir = new File(diretorio);
		if (dir.mkdir())
			b = true;
		return b;
	}
}
```

Pacote com.validacao

Nome do arquivo: Validacao.java

```
package com.validacao;

public enum Validacao {
	NUMERICO, DATA, CPF, CNPJ, DATA_HORA, NOME, OCULTO
}
```

Nome do arquivo: ValidacaoPorCampo.java

```
package com.validacao;

import java.util.HashMap;
import java.util.Map;

public class ValidacaoPorCampo {
	public static Map<String, Validacao> getMapValidacao() {
		Map<String, Validacao> map = new HashMap<String, Validacao>();
		map.put("guiaConsulta.identificacaoGuia.registroANS",
Validacao.NUMERICO);
		map.put("guiaConsulta.beneficiario.validadeCarteira", Validacao.DATA);
		map.put("guiaConsulta.identificacaoGuia.dataEmissaoGuia",
Validacao.DATA);
		map.put("guiaConsulta.identificacaoGuia.numeroGuiaOperadora",
Validacao.NUMERICO);
		map.put("guiaConsulta.identificacaoGuia.numeroGuiaPrestador",
Validacao.NUMERICO);
		map.put("guiaConsulta.beneficiario.numeroCarteira", Validacao.NUMERICO);
		map.put("guiaConsulta.beneficiario.numeroCNS", Validacao.NUMERICO);
		map.put("guiaConsulta.beneficiario.identificadorBeneficiario",
Validacao.NUMERICO);
		map.put("guiaConsulta.dadosContratado.identificacao.CNPJ",
Validacao.NUMERICO);
		map.put("guiaConsulta.dadosContratado.identificacao.CPF",
Validacao.NUMERICO);
```

```
        map.put("guiaConsulta.dadosContratado.identificacao.codigoPrestadorNaOperadora",
Validacao.NUMERICO);
                map.put("guiaConsulta.dadosContratado.enderecoContratado.numero",
Validacao.NUMERICO);

        map.put("guiaConsulta.dadosContratado.enderecoContratado.codigoIBGEMunicipio",
Validacao.NUMERICO);
                map.put("guiaConsulta.dadosContratado.enderecoContratado.cep",
Validacao.NUMERICO);
                map.put("guiaConsulta.dadosContratado.numeroCNES",
Validacao.NUMERICO);

        map.put("guiaConsulta.profissionalExecutante.conselhoProfissional.numeroConselho",
Validacao.NUMERICO);
                map.put("guiaConsulta.profissionalExecutante.cbos", Validacao.NUMERICO);
                map.put("guiaConsulta.dadosAtendimento.dataAtendimento",
Validacao.DATA);
                map.put("guiaConsulta.dadosAtendimento.procedimento.codigoTabela",
Validacao.NUMERICO);

        map.put("guiaConsulta.dadosAtendimento.procedimento.codigoProcedimento",
Validacao.NUMERICO);

                map.put("guiaSP_SADT.identificacaoGuiaSADTSP.dataEmissaoGuia",
Validacao.DATA);
                map.put("guiaSP_SADT.dadosAutorizacao.dataAutorizacao",
Validacao.DATA);
                map.put("guiaSP_SADT.dadosBeneficiario.validadeCarteira",
Validacao.DATA);
                map.put("guiaSP_SADT.procedimentosRealizados.procedimentos.data",
Validacao.DATA);
                map.put("guiaSP_SADT.outrasDespesas.despesa.dataRealizacao",
Validacao.DATA);


        map.put("guiaHonorarioIndividual.identificacaoGuiaHonorarioIndividual.dataEmissaoG
uia", Validacao.DATA);
                map.put("guiaHonorarioIndividual.dadosBeneficiario.validadeCarteira",
Validacao.DATA);

        map.put("guiaHonorarioIndividual.procedimentosExamesRealizados.procedimentoReal
izado.data", Validacao.DATA);


        map.put("guiaResumoInternacao.identificacaoGuiaInternacao.dataEmissaoGuia",
Validacao.DATA);
                map.put("guiaResumoInternacao.dadosBeneficiario.validadeCarteira",
Validacao.DATA);
                map.put("guiaResumoInternacao.dadosAutorizacao.dataAutorizacao",
Validacao.DATA);
                map.put("guiaResumoInternacao.dadosAutorizacao.validadeSenha",
Validacao.DATA);

        map.put("guiaResumoInternacao.procedimentosRealizados.procedimentos.data",
Validacao.DATA);
                map.put("guiaResumoInternacao.outrasDespesas.despesa.dataRealizacao",
Validacao.DATA);
                map.put("guiaSolicInternacao.dataProvavelAdmisHosp", Validacao.DATA);
```

```
        map.put("guiaSolicInternacao.identificacaoGuiaSolicitacaoInternacao.dataEmissaoGui
a", Validacao.DATA);
                map.put("guiaSolicInternacao.dadosBeneficiario.validadeCarteira",
Validacao.DATA);

                return map;
        }
}
```

Arquivos XSD – versão V2_01_03

Nome do arquivo: tissnetV2_01_03.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema
  xmlns:tissnet="http://www.ans.gov.br/padroes/tiss/schemas"
  xmlns:ans="http://www.ans.gov.br/padroes/tiss/schemas"
  elementFormDefault="qualified"
  targetNamespace="http://www.ans.gov.br/padroes/tiss/schemas"
  xmlns="http://www.w3.org/2001/XMLSchema"
>

 <include schemaLocation="tissSimpleTypesV2_01_03.xsd"/>

 <simpleType name="st_host">
  <restriction base="string">
     <whiteSpace value="collapse" />
     <pattern value="[\p{N}\p{L}\./-_]+" />
  </restriction>
 </simpleType>

 <simpleType name="st_wsurl">
  <restriction base="string">
    <whiteSpace value="collapse" />
    <pattern value="[a-z]+://[\p{N}\p{L}\./-_]+\?wsdl" />
  </restriction>
 </simpleType>

 <simpleType name="st_port">
  <restriction base="integer">
    <minExclusive value="0" />
    <maxExclusive value="65536" />
  </restriction>
 </simpleType>

 <complexType name="ct_hostporta">
  <sequence>
     <element name="host" type="tissnet:st_host" />
     <element name="port" type="tissnet:st_port" />
  </sequence>
 </complexType>

 <complexType name="nodo_tissnet">
  <sequence>
    <element name="operadora" type="ans:st_registroANS" />
    <element minOccurs="0" name="pontoAPonto">
     <complexType>
        <choice>
           <element name="socket" type="tissnet:ct_hostporta"/>
```

```
            <element name="service" type="tissnet:st_wsurl"/>
          </choice>
        </complexType>
      </element>
      <element minOccurs="0" name="webServices">
       <complexType>
        <sequence>
         <element minOccurs="0" name="tissCancelaGuia" type="tissnet:st_wsurl" />
         <element minOccurs="0" name="tissLoteGuias" type="tissnet:st_wsurl" />
         <element minOccurs="0" name="tissSolicitacaoAnaliseConta" type="tissnet:st_wsurl"
/>
         <element minOccurs="0" name="tissSolicitacaoDemonstrativoPagamento"
type="tissnet:st_wsurl" />
         <element minOccurs="0" name="tissSolicitacaoProcedimento" type="tissnet:st_wsurl"
/>
         <element minOccurs="0" name="tissSolicitacaoStatusProtocolo" type="tissnet:st_wsurl"
/>
         <element minOccurs="0" name="tissVerificaElegibilidade" type="tissnet:st_wsurl" />
        </sequence>
       </complexType>
      </element>
    </sequence>
   </complexType>

   <element name="definicaoTissNet">
    <complexType>
      <sequence>
        <element name="nodo" type="tissnet:nodo_tissnet" minOccurs="1"
maxOccurs="unbounded"/>
      </sequence>
    </complexType>
   </element>

</schema>
```

Nome do arquivo: tissV2_01_03.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XMLSpy v2006 U (http://www.altova.com) by luiz.vieira (ANS) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ans="http://www.ans.gov.br/padroes/tiss/schemas"
targetNamespace="http://www.ans.gov.br/padroes/tiss/schemas"
elementFormDefault="qualified">
        <!--<include
schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas/tissSimpleTypesV2_01_03.xsd"
/>
        <include
schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas/tissComplexTypesV2_01_03.xs
d"/>
        <include
schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas/tissGuiasV2_01_03.xsd"/>-->
        <include schemaLocation="tissSimpleTypesV2_01_03.xsd"/>
        <include schemaLocation="tissComplexTypesV2_01_03.xsd"/>
        <include schemaLocation="tissGuiasV2_01_03.xsd"/>
        <!-- Definicao do cabecalho das mensagens -->
        <complexType name="cabecalhoTransacao">
                <sequence>
                        <element name="identificacaoTransacao">
                                <complexType>
```

```
                                        <sequence>
                                                <element name="tipoTransacao"
type="ans:st_tipoTransacao"/>
                                                <element name="sequencialTransacao"
type="ans:st_inteiro12"/>
                                                <element name="dataRegistroTransacao"
type="ans:st_data"/>
                                                <element name="horaRegistroTransacao"
type="ans:st_hora"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="falhaNegocio" minOccurs="0">
                                <complexType>
                                        <complexContent>
                                                <extension base="ans:ct_motivoGlosa">
                                                        <sequence>
                                                                <element name="observacao"
type="ans:st_observacao" minOccurs="0"/>
                                                        </sequence>
                                                </extension>
                                        </complexContent>
                                </complexType>
                        </element>
                        <element name="origem">
                                <complexType>
                                        <choice>
                                                <!-- origem de mensagens de prestadoreas
para as operadoras -->
                                                <element
name="codigoPrestadorNaOperadora" type="ans:ct_identificacaoPrestadorExecutante"/>
                                                <!-- origem de mensagens de operadoras para
os prestadores -->
                                                <element name="registroANS"
type="ans:st_registroANS"/>
                                        </choice>
                                </complexType>
                        </element>
                        <element name="destino">
                                <complexType>
                                        <choice>
                                                <!-- destino de mensagens de operadoras
para prestadores: informar código prestador na operadora -->
                                                <element
name="codigoPrestadorNaOperadora" type="ans:ct_identificacaoPrestadorExecutante"/>
                                                <!-- destino de mensagens de prestadores
para as operadoras: informar registro ANS da operadora -->
                                                <element name="registroANS"
type="ans:st_registroANS"/>
                                        </choice>
                                </complexType>
                        </element>
                        <element name="versaoPadrao" type="ans:st_versao"/>
                </sequence>
        </complexType>
        <element name="mensagemTISS">
                <complexType>
                        <sequence>
                                <element name="cabecalho"
type="ans:cabecalhoTransacao"/>
```

```
                        <choice>
                            <element name="operadoraParaPrestador"
minOccurs="0">
                                <complexType>
                                    <choice>
                                        <element
name="protocoloRecebimento" type="ans:ct_protocoloRecebimento" minOccurs="0"/>
                                        <element
name="demonstrativosRetorno" minOccurs="0">
                                            <complexType>
                                                <choice>

        <element name="demonstrativoAnaliseConta"
type="ans:ct_demonstrativoAnaliseConta" minOccurs="0" maxOccurs="30"/>

        <element name="demonstrativoOdontologia" type="ans:ct_demonstrativoOdontologia"
minOccurs="0" maxOccurs="30"/>

        <element name="demonstrativoPagamento" type="ans:ct_demonstrativoPagamento"
minOccurs="0" maxOccurs="30"/>
                                                </choice>
                                            </complexType>
                                        </element>
                                        <element
name="situacaoProtocolo" type="ans:ct_StatusProtocolo" minOccurs="0"/>
                                        <element
name="autorizacaoServico" minOccurs="0" maxOccurs="unbounded">
                                            <complexType>

        <complexContent>

        <extension base="ans:ct_autorizacaoProcedimento">

        <sequence>

            <element name="totalOdonto" minOccurs="0">

                <complexType>

                    <sequence>

                        <element name="totalvalor"
type="ans:st_valorMonetario" minOccurs="0"/>

                        <element name="totalFranquiaCoparticipacao"
type="ans:st_valorMonetario" minOccurs="0"/>

                        <element name="totalquantidadeUS"
type="ans:st_valorMonetario" minOccurs="0"/>

                    </sequence>

                </complexType>

            </element>

            <element name="diariaProrrogacaoAutorizada" minOccurs="0">

                <complexType>
```

```
                        <complexContent>
                            <extension base="ans:ct_diariaProrrogacao">
                                <sequence>
                                    <element
name="tipoAcomodacaoAutorizada" type="ans:st_tipoAcomodacao" minOccurs="0"/>
                                    <element
name="quantidadeAutorizada" type="ans:st_quantidade" minOccurs="0"/>
                                    <element name="motivoGlosa"
type="ans:ct_motivoGlosa" minOccurs="0"/>
                                </sequence>
                            </extension>
                        </complexContent>
                    </complexType>
                </element>
            </sequence>
            </extension>
            </complexContent>
                                                    </complexType>
                                                </element>
                                                <element
name="situacaoAutorizacao" type="ans:ct_situacaoAutorizacao" minOccurs="0"/>
                                                <element
name="respostaElegibilidade" type="ans:ct_retornoElegibilidade" minOccurs="0"/>
                                                <element
name="reciboCancelaGuia" type="ans:ct_reciboCancelaGuia" minOccurs="0"/>
                                            </choice>
                                        </complexType>
                                    </element>
                                    <element name="prestadorParaOperadora"
minOccurs="0">
                                        <complexType>
                                            <choice>
                                                <element name="loteGuias"
type="ans:ct_loteGuias" minOccurs="0"/>
                                                <element
name="solicitacaoDemonstrativoRetorno" type="ans:ct_solicitacaoDemonstrativoPagamento"
minOccurs="0"/>
                                                <element
name="solicitacaoStatusProtocolo" type="ans:ct_solicitacaoStatusProtocolo" minOccurs="0"/>
                                                <element
name="solicitacaoProcedimento" minOccurs="0">
                                                    <complexType>
                                                        <choice>

            <element name="guiaSolicInternacao" type="ans:ct_guiaSolicInternacao"
minOccurs="0"/>
```

```
        <element name="guiaSolicitacaoSADT" type="ans:ct_guiaSolicitacaoSP_SADT"
minOccurs="0"/>

        <element name="guiaSolicitacaoOdontologia" type="ans:ct_guiaSolicitacaoOdonto"
minOccurs="0"/>

        <element name="guiaSolicitacaoProrrogacao"
type="ans:ct_guiaSolicitacaoProrrogacao" minOccurs="0"/>
                                                                    </choice>
                                                                </complexType>
                                                            </element>
                                                            <element
name="solicitaStatusAutorizacao" type="ans:ct_solicitaStatusAutorizacao" minOccurs="0"/>
                                                            <element
name="verificaElegibilidade" type="ans:ct_solicitaElegibilidade" minOccurs="0"/>
                                                            <element name="cancelaGuia"
type="ans:ct_cancelaGuia" minOccurs="0"/>
                                                        </choice>
                                                    </complexType>
                                                </element>
                                            </choice>
                                            <element name="epilogo">
                                                <complexType>
                                                    <sequence>
                                                        <element name="hash" type="string"/>
                                                    </sequence>
                                                </complexType>
                                            </element>
                                        </sequence>
                                    </complexType>
        </element>
</schema>
```

Nome do arquivo: tissGuiasV2_01_03.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XMLSpy v2006 U (http://www.altova.com) by luiz.vieira (ANS) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ans="http://www.ans.gov.br/padroes/tiss/schemas"
targetNamespace="http://www.ans.gov.br/padroes/tiss/schemas"
elementFormDefault="qualified">
        <!--<include
schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas/tissSimpleTypesV2_01_03.xsd"
/>
        <include
schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas/tissComplexTypesV2_01_03.xs
d"/>-->
        <include schemaLocation="tissSimpleTypesV2_01_03.xsd"/>
        <include schemaLocation="tissComplexTypesV2_01_03.xsd"/>
        <complexType name="ct_loteGuias">
                <sequence>
                        <element name="numeroLote" type="ans:st_inteiro12"/>
                        <element name="guias">
                                <complexType>
                                        <complexContent>
                                                <extension base="ans:ct_guias"/>
                                        </complexContent>
                                </complexType>
```

```
                </element>
            </sequence>
        </complexType>
        <!-- Schema com as guias que compõem o  padrão TISS-->
        <complexType name="ct_guias">
            <choice>
                <element name="guiaFaturamento">
                    <complexType>
                        <choice>
                            <element name="guiaConsulta"
type="ans:ct_guiaConsulta" minOccurs="0" maxOccurs="100"/>
                            <element name="guiaSP_SADT"
type="ans:ct_guiaSP_SADT" minOccurs="0" maxOccurs="100"/>
                            <element name="guiaResumoInternacao"
type="ans:ct_guiaResumoInternacao" minOccurs="0" maxOccurs="100"/>
                            <element name="guiaHonorarioIndividual"
type="ans:ct_guiaHonorarioIndividual" minOccurs="0" maxOccurs="100"/>
                            <element name="guiaOdontologia"
type="ans:ct_guiaOdontologia" minOccurs="0" maxOccurs="100"/>
                        </choice>
                    </complexType>
                </element>
                <element name="guiaRevisaoGlosa">
                    <complexType>
                        <choice>
                            <element
name="guiaConsultaReapresentacao" type="ans:ct_guiaConsultaReapresentacao"
minOccurs="0" maxOccurs="100"/>
                            <element
name="guiaSP_SADTReapresentacao" type="ans:ct_guiaSP_SADTReapresentacao"
minOccurs="0" maxOccurs="100"/>
                            <element
name="guiaResumoInternacaoReapresentacao"
type="ans:ct_guiaResumoInternacaoReapresentacao" minOccurs="0" maxOccurs="100"/>
                            <element
name="guiaHonorarioIndividualReapresentacao"
type="ans:ct_guiaHonorarioIndividualReapresentacao" minOccurs="0" maxOccurs="100"/>
                            <element
name="guiaOdontologiaReapresentacao"
type="ans:ct_guiaTratamentoOdontologicoReapresentacao" minOccurs="0"
maxOccurs="100"/>
                        </choice>
                    </complexType>
                </element>
            </choice>
        </complexType>
        <complexType name="ct_guiaHonorarioIndividual">
            <sequence>
                <element name="identificacaoGuiaHonorarioIndividual"
type="ans:ct_cabecalhoGuia"/>
                <element name="numeroGuiaPrincipal" type="ans:st_numeroGuia"/>
                <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                <element name="contratado" type="ans:ct_contratado"/>
                <element name="contratadoExecutante">
                    <complexType>
                        <complexContent>
                            <extension base="ans:ct_contratado">
                                <sequence>
                                    <element
name="identificacaoProfissional" type="ans:ct_profissionalExecutante"/>
```

```
                                                    <element
name="posicaoProfissional" type="ans:st_posicaoProfissao"/>
                                                    <element
name="tipoAcomodacao" type="ans:st_tipoAcomodacao" minOccurs="0"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                        <element name="procedimentosExamesRealizados">
                                            <complexType>
                                                <complexContent>
                                                    <extension
base="ans:ct_procedimentosRealizados">
                                                        <sequence>
                                                            <element
name="totalGeralHonorario" type="ans:st_valorMonetario" minOccurs="0"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                                    </sequence>
                                </complexType>
                                <complexType name="ct_guiaHonorarioIndividualReapresentacao">
                                    <sequence>
                                        <element name="identificacaoGuiaHonorarioIndividual"
type="ans:ct_cabecalhoGuia"/>
                                        <element name="numeroGuiaPrincipal" type="ans:st_numeroGuia"/>
                                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                                        <element name="contratado" type="ans:ct_contratado"/>
                                        <element name="contratadoExecutante">
                                            <complexType>
                                                <complexContent>
                                                    <extension base="ans:ct_contratado">
                                                        <sequence>
                                                            <element
name="identificacaoProfissional" type="ans:ct_profissionalExecutante"/>
                                                            <element
name="posicaoProfissional" type="ans:st_posicaoProfissao"/>
                                                            <element
name="tipoAcomodacao" type="ans:st_tipoAcomodacao" minOccurs="0"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                        <element name="procedimentosExamesRealizados">
                                            <complexType>
                                                <complexContent>
                                                    <extension
base="ans:ct_procedimentoRevisao">
                                                        <sequence>
                                                            <element
name="totalGeralHonorario" type="ans:st_valorMonetario" minOccurs="0"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
```

```
                        </complexType>
                    </element>
                    <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
            </complexType>
            <complexType name="ct_guiaConsulta">
                <sequence>
                    <element name="identificacaoGuia" type="ans:ct_cabecalhoGuia"/>
                    <element name="beneficiario" type="ans:ct_beneficiario"/>
                    <element name="dadosContratado" type="ans:ct_contratado"/>
                    <element name="profissionalExecutante"
type="ans:ct_identificacaoProfissional"/>
                    <element name="hipoteseDiagnostica" minOccurs="0">
                        <complexType>
                            <complexContent>
                                <extension
base="ans:ct_diagnosticoAtendimento">
                                    <sequence>
                                        <element
name="diagnosticoSecundario" type="ans:ct_diagnosticosSecundarios" minOccurs="0"/>
                                    </sequence>
                                </extension>
                            </complexContent>
                        </complexType>
                    </element>
                    <element name="dadosAtendimento">
                        <complexType>
                            <complexContent>
                                <extension
base="ans:ct_atendimentoConsulta"/>
                            </complexContent>
                        </complexType>
                    </element>
                    <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
            </complexType>
            <complexType name="ct_guiaConsultaReapresentacao">
                <sequence>
                    <element name="identificacaoGuia" type="ans:ct_cabecalhoGuia"/>
                    <element name="beneficiario" type="ans:ct_beneficiario"/>
                    <element name="dadosContratado" type="ans:ct_contratado"/>
                    <element name="profissionalExecutante"
type="ans:ct_identificacaoProfissional"/>
                    <element name="hipoteseDiagnostica"
type="ans:ct_diagnosticoAtendimento" minOccurs="0"/>
                    <element name="dadosAtendimento"
type="ans:ct_atendimentoConsultaReapr"/>
                    <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
            </complexType>
            <complexType name="ct_guiaOdontologia">
                <sequence>
                    <element name="cabecalhoGuia"
type="ans:ct_cabecalhoGuiaOdonto"/>
                    <element name="dadosBeneficiario">
                        <complexType>
                            <complexContent>
```

```
                                        <extension base="ans:ct_beneficiario">
                                            <sequence>
                                                <element
name="nomeEmpresa" type="ans:st_descricao40" minOccurs="0"/>
                                                <element
name="numeroTelefone" type="ans:st_numeroTel" minOccurs="0"/>
                                                <element name="nomeTitular"
type="ans:st_nome" minOccurs="0"/>
                                            </sequence>
                                        </extension>
                                    </complexContent>
                                </complexType>
                            </element>
                            <element name="dadosContratadoExecutante">
                                <complexType>
                                    <sequence>
                                        <element name="dadosContratado">
                                            <complexType>
                                                <complexContent>
                                                    <extension
base="ans:ct_contratado">
                                                        <sequence>

        <element name="conselhoProfissional" type="ans:ct_conselhoProfissional"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                        <element name="dadosExecutante"
type="ans:ct_profissionalExecutante" minOccurs="0"/>
                                    </sequence>
                                </complexType>
                            </element>
                            <element name="procedimentosExecutados">
                                <complexType>
                                    <sequence>
                                        <element name="procedimentoExecutado"
maxOccurs="unbounded">
                                            <complexType>
                                                <complexContent>
                                                    <extension
base="ans:ct_procedimentoOdontologia">
                                                        <sequence>

        <element name="dataRealizacao" type="ans:st_data"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                        <element name="quantidadeTotalUS"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        <element name="valorTotal"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        <element name="totalFranquiaCoparticipacao"
type="ans:st_valorMonetario" minOccurs="0"/>
                                    </sequence>
                                </complexType>
                            </element>
```

```
                        <element name="outrasDespesas" type="ans:ct_outrasDespesas"
minOccurs="0"/>
                        <element name="dadosAtendimento"
type="ans:ct_atendimentoOdontologia" minOccurs="0"/>
                        <element name="tipoFaturamento" type="ans:st_tipoFaturamento"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaResumoInternacao">
                <sequence>
                        <element name="identificacaoGuiaInternacao"
type="ans:ct_cabecalhoGuia"/>
                        <element name="numeroGuiaSolicitacao" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="identificacaoExecutante" type="ans:ct_contratado"/>
                        <element name="dadosAutorizacao" type="ans:ct_autorizacaoGuia"
minOccurs="0"/>
                        <element name="caraterInternacao"
type="ans:st_eletivaEmergencia"/>
                        <element name="acomodacao" type="ans:st_tipoAcomodacao"/>
                        <element name="dataHoraInternacao" type="ans:st_dataHora"/>
                        <element name="dataHoraSaidaInternacao" type="ans:st_dataHora"
minOccurs="0"/>
                        <element name="tipoInternacao" type="ans:st_tipoInternacao"/>
                        <element name="regimeInternacao" type="ans:st_regimeInternacao"/>
                        <element name="internacaoObstetrica"
type="ans:ct_internacaoObstetrica" minOccurs="0"/>
                        <element name="obitoNeonatal" type="ans:ct_obitoNeonatal"
minOccurs="0"/>
                        <element name="diagnosticosSaidaInternacao"
type="ans:ct_diagnosticoSaidaInternacao"/>
                        <element name="procedimentosRealizados"
type="ans:ct_procedimentoRealizadoEquipe" minOccurs="0"/>
                        <element name="OPMUtilizadas" type="ans:ct_OPMUtilizada"
minOccurs="0"/>
                        <element name="outrasDespesas" type="ans:ct_outrasDespesas"
minOccurs="0"/>
                        <element name="tipoFaturamento" type="ans:st_tipoFaturamento"/>
                        <element name="valorTotal" type="ans:ct_valorTotalServicos"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaResumoInternacaoReapresentacao">
                <sequence>
                        <element name="identificacaoGuiaInternacao"
type="ans:ct_cabecalhoGuia"/>
                        <element name="numeroGuiaSolicitacao" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="identificacaoExecutante" type="ans:ct_contratado"/>
                        <element name="dadosAutorizacao" type="ans:ct_autorizacaoGuia"
minOccurs="0"/>
                        <element name="caraterInternacao"
type="ans:st_eletivaEmergencia"/>
                        <element name="acomodacao" type="ans:st_tipoAcomodacao"/>
```

```
                        <element name="dataHoraInternacao" type="ans:st_dataHora"/>
                        <element name="dataHoraSaidaInternacao" type="ans:st_dataHora"
minOccurs="0"/>
                        <element name="tipoInternacao" type="ans:st_tipoInternacao"/>
                        <element name="regimeInternacao" type="ans:st_regimeInternacao"/>
                        <element name="internacaoObstetrica"
type="ans:ct_internacaoObstetrica" minOccurs="0"/>
                        <element name="obitoNeonatal" type="ans:ct_obitoNeonatal"
minOccurs="0"/>
                        <element name="diagnosticosSaidaInternacao"
type="ans:ct_diagnosticoSaidaInternacao"/>
                        <element name="procedimentosRealizados"
type="ans:ct_procedimentoRevisao" minOccurs="0"/>
                        <element name="OPMUtilizadas"
type="ans:ct_OPMUtilizadasRevisao" minOccurs="0"/>
                        <element name="outrasDespesas"
type="ans:ct_outrasDespesasRevisao" minOccurs="0"/>
                        <element name="tipoFaturamento" type="ans:st_tipoFaturamento"/>
                        <element name="valorTotal" type="ans:ct_valorTotalServicos"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaSolicInternacao">
                <sequence>
                        <element name="identificacaoGuiaSolicitacaoInternacao"
type="ans:ct_cabecalhoSolicitacao"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="dadosSolicitante" type="ans:ct_solicitante"/>
                        <element name="prestadorSolicitado" minOccurs="0">
                                <complexType>
                                        <complexContent>
                                                <extension
base="ans:ct_identificacaoPrestador">
                                                        <sequence>
                                                                <element
name="nomePrestador" type="ans:st_nome"/>
                                                        </sequence>
                                                </extension>
                                        </complexContent>
                                </complexType>
                        </element>
                        <element name="caraterInternacao"
type="ans:st_eletivaEmergencia"/>
                        <element name="tipoInternacao" type="ans:st_tipoInternacao"/>
                        <element name="indicacaoClinica" type="ans:st_indicacaoClinica"/>
                        <element name="regimeInternacao" type="ans:st_regimeInternacao"/>
                        <element name="hipotesesDiagnosticas"
type="ans:ct_hipoteseDiagnostica"/>
                        <element name="procedimentosExamesSolicitados"
type="ans:ct_procedimentosSolicitados"/>
                        <element name="OPMsSolicitadas" type="ans:ct_OPMSolicitadas"
minOccurs="0"/>
                        <element name="diasSolicitados" type="integer"/>
                        <element name="dataProvavelAdmisHosp" type="ans:st_data"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
```

```
        </complexType>
        <complexType name="ct_guiaSolicitacaoOdonto">
                <sequence>
                        <element name="cabecalhoGuia"
type="ans:ct_cabecalhoGuiaOdonto"/>
                        <element name="dadosBeneficiario">
                                <complexType>
                                        <complexContent>
                                                <extension base="ans:ct_beneficiario">
                                                        <sequence>
                                                                <element
name="nomeEmpresa" type="ans:st_descricao40" minOccurs="0"/>
                                                                <element
name="numeroTelefone" type="ans:st_numeroTel" minOccurs="0"/>
                                                                <element name="nomeTitular"
type="ans:st_nome" minOccurs="0"/>
                                                        </sequence>
                                                </extension>
                                        </complexContent>
                                </complexType>
                        </element>
                        <element name="dadosSolicitante">
                                <complexType>
                                        <sequence>
                                                <element name="dadosContratado">
                                                        <complexType>
                                                                <complexContent>
                                                                        <extension
base="ans:ct_contratado">
                                                                                <sequence>

        <element name="conselhoProfissional" type="ans:ct_conselhoProfissional"/>
                                                                                </sequence>
                                                                        </extension>
                                                                </complexContent>
                                                        </complexType>
                                                </element>
                                                <element name="dadosExecutante"
minOccurs="0">
                                                        <complexType>
                                                                <complexContent>
                                                                        <extension
base="ans:ct_profissionalExecutante"/>
                                                                </complexContent>
                                                        </complexType>
                                                </element>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="situacaoInicial" type="ans:ct_situacaoInicial"
minOccurs="0"/>
                        <element name="planoTratamento">
                                <complexType>
                                        <sequence>
                                                <element name="procedimentoSolicitado"
type="ans:ct_procedimentoOdontologia" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
```

```
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaSolicitacaoProrrogacao">
                <sequence>
                        <element name="identificacaoProrrogacao"
type="ans:ct_cabecalhoGuia"/>
                        <element name="identificacaoGuiaReferencia"
type="ans:ct_cabecalhoGuia"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="dadosSolicitante" type="ans:ct_solicitante"/>
                        <element name="indicacaoClinica" type="ans:st_indicacaoClinica"/>
                        <element name="responsavelAutorizacao" type="ans:st_nome"
minOccurs="0"/>
                        <element name="procedimentosExamesSolicitados"
type="ans:ct_procedimentosSolicitados" minOccurs="0"/>
                        <element name="diarias" type="ans:ct_diariaProrrogacao"
minOccurs="0"/>
                        <element name="OPMsSolicitadas" type="ans:ct_OPMSolicitadas"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaSolicitacaoSP_SADT">
                <sequence>
                        <element name="identificacaoGuiaSolicitacaoSADTSP"
type="ans:ct_cabecalhoSolicitacao"/>
                        <element name="numeroGuiaPrincipal" type="integer"
minOccurs="0"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="dadosSolicitante" type="ans:ct_solicitante"/>
                        <element name="caraterSolicitacao"
type="ans:st_eletivaEmergencia"/>
                        <element name="dataHoraSolicitacao" type="ans:st_dataHora"/>
                        <element name="Diagnostico" type="ans:ct_diagnostico"
nillable="true" minOccurs="0"/>
                        <element name="indicacaoClinica" type="ans:st_indicacaoClinica"
nillable="true" minOccurs="0"/>
                        <element name="procedimentosSolicitados"
type="ans:ct_procedimentosSolicitados"/>
                        <element name="opmSolicitada" type="ans:ct_OPMSolicitadas"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaSP_SADT">
                <sequence>
                        <element name="identificacaoGuiaSADTSP"
type="ans:ct_cabecalhoGuia"/>
                        <element name="numeroGuiaPrincipal" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="dadosAutorizacao" type="ans:ct_autorizacaoGuia"
minOccurs="0"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="dadosSolicitante" type="ans:ct_solicitante"/>
                        <element name="prestadorExecutante">
                                <complexType>
```

```
                                        <complexContent>
                                            <extension base="ans:ct_contratado">
                                                <sequence>
                                                    <element
name="profissionalExecutanteCompl" minOccurs="0">
                                                        <complexType>

        <complexContent>

        <extension base="ans:ct_profissionalExecutante">

        <sequence>

                <element name="codigoProfissionalCompl"
type="ans:ct_identificacaoProfissionalCompl" minOccurs="0"/>

        </sequence>

        </extension>

        </complexContent>
                                                        </complexType>
                                                    </element>
                                                </sequence>
                                            </extension>
                                        </complexContent>
                                    </complexType>
                                </element>
                                <element name="indicacaoClinica" type="ans:st_indicacaoClinica"
minOccurs="0"/>
                                <element name="caraterAtendimento"
type="ans:st_eletivaEmergencia"/>
                                <element name="dataHoraAtendimento" type="ans:st_dataHora"
minOccurs="0"/>
                                <element name="diagnosticoAtendimento"
type="ans:ct_diagnosticoAtendimento" minOccurs="0"/>
                                <element name="tipoSaida" type="ans:st_tipoSaidaGuiaSADT"/>
                                <element name="tipoAtendimento" type="ans:st_tipoAtendimento"/>
                                <element name="procedimentosRealizados"
type="ans:ct_procedimentoRealizadoEquipe" minOccurs="0"/>
                                <element name="outrasDespesas" type="ans:ct_outrasDespesas"
minOccurs="0"/>
                                <element name="OPMUtilizada" type="ans:ct_OPMUtilizada"
minOccurs="0"/>
                                <element name="valorTotal" type="ans:ct_valorTotalServicos"
minOccurs="0"/>
                                <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                        </sequence>
        </complexType>
        <complexType name="ct_guiaSP_SADTReapresentacao">
                <sequence>
                        <element name="identificacaoGuiaSADTSP"
type="ans:ct_cabecalhoGuia"/>
                        <element name="numeroGuiaPrincipal" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="dadosAutorizacao" type="ans:ct_autorizacaoGuia"
minOccurs="0"/>
                        <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        <element name="dadosSolicitante" type="ans:ct_solicitante"/>
```

```
                        <element name="prestadorExecutante">
                            <complexType>
                                <complexContent>
                                    <extension base="ans:ct_contratado">
                                        <sequence>
                                            <element
name="profissionalExecutanteCompl" minOccurs="0">
                                                <complexType>

        <complexContent>

        <extension base="ans:ct_profissionalExecutante">

        <sequence>

                <element name="codigoProfissionalCompl"
type="ans:ct_identificacaoProfissionalCompl" minOccurs="0"/>

        </sequence>

        </extension>

        </complexContent>
                                                </complexType>
                                            </element>
                                        </sequence>
                                    </extension>
                                </complexContent>
                            </complexType>
                        </element>
                        <element name="indicacaoClinica" type="ans:st_indicacaoClinica"
minOccurs="0"/>
                        <element name="caraterAtendimento"
type="ans:st_eletivaEmergencia"/>
                        <element name="dataHoraAtendimento" type="ans:st_dataHora"
minOccurs="0"/>
                        <element name="diagnosticoGuia"
type="ans:ct_diagnosticoAtendimento" minOccurs="0"/>
                        <element name="tipoSaida" type="ans:st_tipoSaidaGuiaSADT"/>
                        <element name="tipoAtendimento" type="ans:st_tipoAtendimento"/>
                        <element name="procedimentosRealizados"
type="ans:ct_procedimentoRevisao" minOccurs="0"/>
                        <element name="outrasDespesas"
type="ans:ct_outrasDespesasRevisao" minOccurs="0"/>
                        <element name="OPMUtilizada" type="ans:ct_OPMUtilizadasRevisao"
minOccurs="0"/>
                        <element name="valorTotal" type="ans:ct_valorTotalServicos"
minOccurs="0"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_guiaTratamentoOdontologicoReapresentacao">
                <sequence>
                        <element name="cabecalhoGuia"
type="ans:ct_cabecalhoGuiaOdonto"/>
                        <element name="dadosBeneficiario">
                            <complexType>
                                <complexContent>
                                    <extension base="ans:ct_beneficiario">
```

```
                                                    <sequence>
                                                        <element
name="nomeEmpresa" type="ans:st_descricao40" minOccurs="0"/>
                                                        <element
name="numeroTelefone" type="ans:st_numeroTel" minOccurs="0"/>
                                                        <element name="nomeTitular"
type="ans:st_nome" minOccurs="0"/>
                                                    </sequence>
                                                </extension>
                                            </complexContent>
                                        </complexType>
                                    </element>
                                    <element name="dadosContratadoExecutante">
                                        <complexType>
                                            <sequence>
                                                <element name="dadosContratado">
                                                    <complexType>
                                                        <complexContent>
                                                            <extension
base="ans:ct_contratado">
                                                                <sequence>

        <element name="conselhoProfissional" type="ans:ct_conselhoProfissional"/>
                                                                </sequence>
                                                            </extension>
                                                        </complexContent>
                                                    </complexType>
                                                </element>
                                                <element name="dadosExecutante"
type="ans:ct_profissionalExecutante" minOccurs="0"/>
                                            </sequence>
                                        </complexType>
                                    </element>
                                    <element name="outrasDespesas" type="ans:ct_outrasDespesas"
minOccurs="0"/>
                                    <element name="dadosAtendimento"
type="ans:ct_atendimentoOdontologia" minOccurs="0"/>
                                    <element name="procedimentosExecutados">
                                        <complexType>
                                            <sequence>
                                                <element name="procedimentoExecutado"
type="ans:ct_procedimentoOdontoRevisao" maxOccurs="unbounded"/>
                                            </sequence>
                                        </complexType>
                                    </element>
                                    <element name="tipoFaturamento" type="ans:st_tipoFaturamento"
minOccurs="0"/>
                                    <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                                </sequence>
                            </complexType>
</schema>
```

Nome do arquivo: tissComplexTypesV2_01_03.xsd

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ans="http://www.ans.gov.br/padroes/tiss/schemas"
targetNamespace="http://www.ans.gov.br/padroes/tiss/schemas"
elementFormDefault="qualified">
```

```
	<!-- -->
	<!--<include
schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas/tissSimpleTypesV2_01_03.xsd"
/>-->
	<include schemaLocation="tissSimpleTypesV2_01_03.xsd"/>
	<complexType name="ct_atendimentoConsulta">
		<sequence>
			<element name="dataAtendimento" type="ans:st_data"/>
			<element name="procedimento">
				<complexType>
					<all>
						<element name="codigoTabela"
type="ans:st_tabela"/>
						<element name="codigoProcedimento"
type="ans:st_codigoTabela"/>
					</all>
				</complexType>
			</element>
			<element name="tipoConsulta" type="ans:st_tipoConsulta"/>
			<element name="tipoSaida" type="ans:st_tipoSaidaGuiaConsulta"/>
		</sequence>
	</complexType>
	<complexType name="ct_atendimentoConsultaReapr">
		<sequence>
			<element name="procedimento">
				<complexType>
					<all>
						<element name="codigoTabela"
type="ans:st_tabela"/>
						<element name="codigoProcedimento"
type="ans:st_codigoTabela"/>
						<element name="justificativa"
type="ans:st_justificativa"/>
					</all>
				</complexType>
			</element>
			<element name="tipoConsulta" type="ans:st_tipoConsulta"/>
			<element name="tipoSaida" type="ans:st_tipoSaidaGuiaConsulta"/>
			<element name="dataAtendimento" type="ans:st_data"/>
		</sequence>
	</complexType>
	<complexType name="ct_atendimentoOdontologia">
		<sequence>
			<element name="dataTerminoTratamento" type="ans:st_data"/>
			<element name="tipoAtendimento"
type="ans:st_atendimentoOdonto"/>
		</sequence>
	</complexType>
	<complexType name="ct_autorizacaoGuia">
		<sequence>
			<element name="dataAutorizacao" type="ans:st_data"
minOccurs="0"/>
			<element name="senhaAutorizacao" type="ans:st_senhaAutorizacao"
minOccurs="0"/>
			<element name="validadeSenha" type="ans:st_data" minOccurs="0"/>
		</sequence>
	</complexType>
	<complexType name="ct_autorizacaoGuiaInternacao">
		<complexContent>
			<extension base="ans:ct_autorizacaoGuia">
```

```
                                <sequence>
                                        <element name="diasAutorizado"
type="ans:st_numeroInteiro" minOccurs="0"/>
                                        <element name="dataProvavelAdmisHosp"
type="ans:st_data" minOccurs="0"/>
                                        <element name="tipoAcomodacao"
type="ans:st_tipoAcomodacao" minOccurs="0"/>
                                </sequence>
                        </extension>
                </complexContent>
        </complexType>
        <complexType name="ct_autorizacaoProcedimento">
                <sequence>
                        <element name="identificacaoAutorizacao"
type="ans:ct_cabecalhoGuia"/>
                        <element name="statusSolicitacao" type="ans:st_statusSolicitacao"
minOccurs="0"/>
                        <element name="motivoNegativa" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element name="motivoGlosa"
type="ans:ct_motivoGlosa" minOccurs="0" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="beneficiario" type="ans:ct_beneficiario"/>
                        <element name="prestadorAutorizado" type="ans:ct_contratado"/>
                        <element name="dadosAutorizacao"
type="ans:ct_autorizacaoGuiaInternacao" minOccurs="0"/>
                        <element name="procedimentos" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element name="procedimento"
type="ans:ct_itemSolicitacao" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_beneficiario">
                <sequence>
                        <element name="numeroCarteira" type="ans:st_descricao20"/>
                        <element name="nomeBeneficiario" type="ans:st_nome"/>
                        <element name="nomePlano" type="ans:st_descricao40"/>
                        <element name="validadeCarteira" type="ans:st_data"
minOccurs="0"/>
                        <element name="numeroCNS" type="ans:st_descricao15"
minOccurs="0"/>
                        <element name="identificadorBeneficiario" type="base64Binary"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_cabecalhoGuia">
                <sequence>
                        <element name="registroANS" type="ans:st_registroANS"/>
                        <element name="dataEmissaoGuia" type="ans:st_data"/>
                        <element name="numeroGuiaPrestador" type="ans:st_numeroGuia"/>
                        <element name="numeroGuiaOperadora" type="ans:st_numeroGuia"
minOccurs="0"/>
                </sequence>
```

```
        </complexType>
        <complexType name="ct_cabecalhoGuiaOdonto">
                <sequence>
                        <element name="registroANS" type="ans:st_registroANS"/>
                        <element name="numeroGuiaPrestador" type="ans:st_numeroGuia"/>
                        <element name="numeroGuiaOperadora" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="dataEmissaoGuia" type="ans:st_data"
minOccurs="0"/>
                        <element name="numeroGuiaPrincipal" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <sequence>
                                <element name="dataAutorizacao" type="ans:st_data"
minOccurs="0"/>
                                <element name="senhaAutorizacao"
type="ans:st_senhaAutorizacao" minOccurs="0"/>
                                <element name="validadeSenha" type="ans:st_data"
minOccurs="0"/>
                        </sequence>
                </sequence>
        </complexType>
        <complexType name="ct_cabecalhoDemonstrativo">
                <sequence>
                        <element name="numeroDemonstrativo" type="ans:st_descricao20"/>
                        <element name="dataEmissao" type="ans:st_data"/>
                </sequence>
        </complexType>
        <complexType name="ct_cabecalhoSolicitacao">
                <sequence>
                        <element name="registroANS" type="ans:st_registroANS"/>
                        <element name="dataEmissaoGuia" type="ans:st_data"
minOccurs="0"/>
                        <element name="numeroGuiaOperadora" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="numeroGuiaPrestador" type="ans:st_numeroGuia"/>
                </sequence>
        </complexType>
        <complexType name="ct_cancelaGuia">
                <sequence>
                        <element name="dadosPrestador" type="ans:ct_contratado"/>
                        <element name="guia">
                                <complexType>
                                        <sequence>
                                                <element name="dadosGuia"
type="ans:ct_cabecalhoGuia" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_contratado">
                <sequence>
                        <element name="identificacao"
type="ans:ct_identificacaoPrestadorExecutante"/>
                        <element name="nomeContratado" type="ans:st_nome"/>
                        <element name="enderecoContratado" type="ans:ct_endereco"
minOccurs="0"/>
```

```
                    <element name="numeroCNES" type="ans:st_CNES"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_contratadoSolicitante">
                <sequence>
                    <element name="identificacao" type="ans:ct_identificacaoPrestador"/>
                    <element name="nomeContratado" type="ans:st_nome"/>
                    <element name="enderecoContratado" type="ans:ct_endereco"
minOccurs="0"/>
                    <element name="numeroCNES" type="ans:st_CNES"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_conselhoProfissional">
                <sequence>
                    <element name="siglaConselho" type="ans:st_conselhoProfissional"/>
                    <element name="numeroConselho" type="ans:st_descricao15"/>
                    <element name="ufConselho" type="ans:st_UF"/>
                </sequence>
        </complexType>
        <complexType name="ct_creditoOdonto">
                <sequence>
                    <element name="valorDesconto" type="ans:st_valorMonetario"/>
                    <element name="descricao" type="ans:st_descricao40"/>
                </sequence>
        </complexType>
        <complexType name="ct_dadoBancario">
                <sequence>
                    <element name="dataPagamento" type="ans:st_data"/>
                    <element name="formaPagamento" type="ans:st_formaPagamento"/>
                    <element name="banco" type="ans:st_banco" minOccurs="0"/>
                    <element name="agencia" type="ans:st_agencia" minOccurs="0"/>
                    <element name="contaCorrente" type="ans:st_conta" minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_dadosConta">
                <sequence>
                    <element name="numeroFatura" type="ans:st_descricao12"
minOccurs="0"/>
                    <element name="lote">
                        <complexType>
                            <sequence>
                                <element name="dadosLote"
type="ans:ct_dadosLote" maxOccurs="unbounded"/>
                            </sequence>
                        </complexType>
                    </element>
                    <element name="valorProcessadoFatura"
type="ans:st_valorMonetario"/>
                    <element name="valorLiberadoFatura" type="ans:st_valorMonetario"/>
                    <element name="valorGlosaFatura" type="ans:st_valorMonetario"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_dadosGuia">
                <sequence>
                    <element name="numeroGuiaPrestador" type="ans:st_numeroGuia"/>
                    <element name="numeroGuiaOperadora" type="ans:st_numeroGuia"
minOccurs="0"/>
```

```
                        <element name="beneficiario" type="ans:ct_beneficiario"/>
                        <element name="dataRealizacao" type="ans:st_data"/>
                        <element name="valorProcessadoGuia" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="valorLiberadoGuia" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="valorGlosaGuia" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="procedimentos">
                                <complexType>
                                        <sequence>
                                                <element name="dadosProcedimento"
type="ans:ct_dadosProcedimento" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="codigoGlosaGuia" type="ans:ct_motivoGlosa"
minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
        </complexType>
        <complexType name="ct_dadosLote">
                <sequence>
                        <element name="numeroLote" type="ans:st_descricao12"/>
                        <element name="dataEnvioLote" type="ans:st_data"/>
                        <element name="numeroProtocolo" type="ans:st_protocolo"/>
                        <element name="valorProtocolo" type="ans:st_valorMonetario"/>
                        <element name="valorGlosaProtocolo" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="codigoGlosaProtocolo" type="ans:st_tipoGlosa"
minOccurs="0" maxOccurs="unbounded"/>
                        <element name="guia">
                                <complexType>
                                        <sequence>
                                                <element name="dadosGuia"
type="ans:ct_dadosGuia" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_dadosPagamento">
                <sequence>
                        <element name="dataPagamento" type="ans:st_data"/>
                        <element name="loteOdonto">
                                <complexType>
                                        <complexContent>
                                                <extension base="ans:ct_loteOdonto">
                                                        <sequence>
                                                                <element
name="totalGeralLote" type="ans:ct_totalDemoOdonto"/>
                                                        </sequence>
                                                </extension>
                                        </complexContent>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_dadosProcedimento">
                <sequence>
                        <element name="procedimento" type="ans:ct_tabela"/>
```

```
                        <element name="quantidadeExecutada"
type="ans:st_quantidadeProcedimento"/>
                        <element name="valorProcessado" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="valorLiberado" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="grauParticipacao" type="ans:st_posicaoProfissao"
minOccurs="0"/>
                        <element name="relacaoGlosa" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element name="tipoGlosa"
type="ans:ct_motivoGlosa" maxOccurs="unbounded"/>
                                                <element name="valorGlosa"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_dadosResumo">
                <sequence>
                        <element name="numeroFatura" type="ans:st_descricao12"
minOccurs="0"/>
                        <element name="numeroLote" type="ans:st_descricao12"/>
                        <element name="dataEnvioLote" type="ans:st_data"/>
                        <element name="numeroProtocolo" type="ans:st_protocolo"/>
                        <element name="valorInformado" type="ans:st_valorMonetario"/>
                        <element name="valorProcessado" type="ans:st_valorMonetario"/>
                        <element name="valorLiberado" type="ans:st_valorMonetario"/>
                        <element name="valorGlosa" type="ans:st_valorMonetario"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_demonstrativoAnaliseConta">
                <sequence>
                        <element name="identificacaoOperadora"
type="ans:ct_identificacaoOperadora"/>
                        <element name="cabecalhoDemonstrativo"
type="ans:ct_cabecalhoDemonstrativo"/>
                        <element name="dadosPrestador" type="ans:ct_contratado"/>
                        <element name="fatura">
                                <complexType>
                                        <sequence>
                                                <element name="dadosFatura"
type="ans:ct_dadosConta"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="valorProcessadoGeral"
type="ans:st_valorMonetario"/>
                        <element name="valorLiberadoGeral" type="ans:st_valorMonetario"/>
                        <element name="valorGlosaGeral" type="ans:st_valorMonetario"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_demonstrativoPagamento">
                <sequence>
                        <element name="cabecalhoDemonstrativo"
type="ans:ct_cabecalhoDemonstrativo"/>
```

```
                        <element name="identificacaoOperadora"
type="ans:ct_identificacaoOperadora"/>
                        <element name="dadosPrestador" type="ans:ct_contratado"/>
                        <element name="dadosPagamento" type="ans:ct_dadoBancario"/>
                        <element name="dadosResumo">
                                <complexType>
                                        <sequence>
                                                <element name="detalheResumo"
type="ans:ct_dadosResumo" maxOccurs="unbounded"/>
                                                <element name="totalGeralInformado"
type="ans:st_valorMonetario"/>
                                                <element name="TotalGeralProcessado"
type="ans:st_valorMonetario"/>
                                                <element name="TotalGeralGlosa"
type="ans:st_valorMonetario" minOccurs="0"/>
                                                <element name="TotalGeralLiberado"
type="ans:st_valorMonetario"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="valorTotal" type="ans:st_valorMonetario"/>
                        <element name="demaisDescontos" type="ans:ct_descontos"
minOccurs="0"/>
                        <element name="valorFinalLiberado" type="ans:st_valorMonetario"/>
                </sequence>
        </complexType>
        <complexType name="ct_demonstrativoOdontologia">
                <sequence>
                        <element name="cabecalhoDemonstrativo"
type="ans:ct_cabecalhoDemonstrativo"/>
                        <element name="identificacaoOperadora">
                                <complexType>
                                        <complexContent>
                                                <extension
base="ans:ct_identificacaoOperadora">
                                                        <sequence>
                                                                <element
name="periodoProcessamento" type="ans:ct_periodoProcessamento"/>
                                                        </sequence>
                                                </extension>
                                        </complexContent>
                                </complexType>
                        </element>
                        <element name="dadosPrestador">
                                <complexType>
                                        <sequence>
                                                <element
name="codigoPrestadorNaOperadora" type="ans:st_codigoPrestadorNaOperadora"
minOccurs="0"/>
                                                <element name="identificadorPrestador"
type="ans:ct_identificadorPrestOdontoDemo"/>
                                                <element name="nomeContratado"
type="ans:st_nome"/>
                                                <element name="numeroCNES"
type="ans:st_CNES" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="dadosPagamento">
                                <complexType>
```

```
                                        <complexContent>
                                                <extension base="ans:ct_dadosPagamento">
                                                        <sequence>
                                                                <element name="totalGeral"
type="ans:ct_totalDemoOdonto"/>
                                                        </sequence>
                                                </extension>
                                        </complexContent>
                                </complexType>
                        </element>
                        <element name="debitosCreditosTributaveis">
                                <complexType>
                                        <sequence>
                                                <element name="descontos"
type="ans:ct_descontoOdonto" minOccurs="0" maxOccurs="unbounded"/>
                                                <element name="creditos"
type="ans:ct_creditoOdonto" minOccurs="0" maxOccurs="unbounded"/>
                                                <element name="valorTotalTributavel"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="demaisImpostos">
                                <complexType>
                                        <sequence>
                                                <element name="descontos"
type="ans:ct_descontoOdonto" minOccurs="0" maxOccurs="unbounded"/>
                                                <element name="valorTotalImpostos"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="debitosCreditosNaoTributaveis">
                                <complexType>
                                        <sequence>
                                                <element name="descontos"
type="ans:ct_descontoOdonto" minOccurs="0" maxOccurs="unbounded"/>
                                                <element name="creditos"
type="ans:ct_creditoOdonto" minOccurs="0" maxOccurs="unbounded"/>
                                                <element name="valorTotalNaoTributavel"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="valorFinalReceber" type="ans:st_valorMonetario"/>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_descontoOdonto">
                <sequence>
                        <element name="descricao" type="ans:st_descricao40"/>
                        <element name="valorDesconto">
                                <simpleType>
                                        <restriction base="ans:st_valorMonetario">
                                                </restriction>
                                </simpleType>
                        </element>
                </sequence>
        </complexType>
```

```
        <complexType name="ct_detalheGuiaOdonto">
                <sequence>
                        <element name="numeroGuiaPrestador" type="ans:st_numeroGuia"/>
                        <element name="numeroGuiaOperadora" type="ans:st_numeroGuia"
minOccurs="0"/>
                        <element name="codigoBeneficiario"
type="ans:st_codigoBeneficiario"/>
                        <element name="NomeBeneficiario" type="ans:st_nome"/>
                        <element name="procedimentoRealizado">
                                <complexType>
                                        <sequence>
                                                <element name="procedimentosOdonto"
type="ans:ct_procedimentoOdontoRealizado" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="valorTotalGlosaGuia" type="ans:st_valorMonetario"/>
                        <element name="valorTotalLiberadoGuia"
type="ans:st_valorMonetario"/>
                        <element name="valorTotalProcessadoGuia"
type="ans:st_valorMonetario"/>
                </sequence>
        </complexType>
        <complexType name="ct_detalheProtocolo">
                <sequence>
                        <element name="numeroProtocolo" type="ans:st_protocolo"/>
                        <element name="valorProtocolo" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="codigoGlosaProtocolo" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element name="motivoGlosa"
type="ans:ct_motivoGlosa" maxOccurs="unbounded"/>
                                                <element name="valorGlosaProtocolo"
type="ans:st_valorMonetario"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="guias">
                                <complexType>
                                        <sequence>
                                                <element name="dadosGuia"
type="ans:ct_dadosGuia" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_denteRegiao">
                <choice>
                        <element name="dente" type="ans:st_dente"/>
                        <element name="regiao" type="ans:st_regiao"/>
                </choice>
        </complexType>
        <complexType name="ct_descontos">
                <sequence>
                        <element name="itemDesconto" maxOccurs="unbounded">
                                <complexType>
                                        <sequence>
```

```
                                        <element name="valorDesconto"
type="ans:st_valorMonetario"/>
                                        <element name="descricao"
type="ans:st_descricao40"/>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
                </complexType>
                <complexType name="ct_diagnostico">
                        <sequence>
                            <element name="nomeTabela" type="ans:st_tabelasDiagnostico"/>
                            <element name="codigoDiagnostico"
type="ans:st_codigoDiagnostico"/>
                            <element name="descricaoDiagnostico" type="ans:st_nome"
nillable="true" minOccurs="0"/>
                        </sequence>
                </complexType>
                <complexType name="ct_diagnosticoAtendimento">
                        <sequence>
                            <element name="CID" type="ans:ct_diagnostico" minOccurs="0"/>
                            <element name="tipoDoenca" type="ans:st_tipoDoenca" nillable="true"
minOccurs="0"/>
                            <element name="tempoReferidoEvolucaoDoenca"
type="ans:ct_tempoEvolucaoDoenca" nillable="true" minOccurs="0"/>
                            <element name="indicadorAcidente" type="ans:st_indicadorAcidente"
minOccurs="0"/>
                        </sequence>
                </complexType>
                <complexType name="ct_diagnosticoGuia">
                        <sequence>
                            <element name="CID" type="ans:ct_diagnostico"/>
                            <element name="tipoDoenca" type="ans:st_tipoDoenca" nillable="true"
minOccurs="0"/>
                            <element name="tempoReferidoEvolucaoDoenca"
type="ans:ct_tempoEvolucaoDoenca" nillable="true" minOccurs="0"/>
                            <element name="indicadorAcidente" type="ans:st_indicadorAcidente"
minOccurs="0"/>
                        </sequence>
                </complexType>
                <complexType name="ct_diagnosticoSaidaInternacao">
                        <sequence>
                            <element name="diagnosticoPrincipal" type="ans:ct_diagnostico"/>
                            <element name="indicadorAcidente" type="ans:st_indicadorAcidente"
minOccurs="0"/>
                            <element name="diagnosticosSecundarios"
type="ans:ct_diagnosticosSecundarios" minOccurs="0"/>
                            <element name="motivoSaidaInternacao" type="ans:st_motivoSaida"/>
                            <element name="obito" type="ans:ct_obitoInternacao" minOccurs="0"
maxOccurs="unbounded"/>
                        </sequence>
                </complexType>
                <complexType name="ct_diagnosticosSecundarios">
                        <sequence>
                            <element name="CID" type="ans:ct_diagnostico" maxOccurs="3"/>
                        </sequence>
                </complexType>
                <complexType name="ct_diariaProrrogacaoAutorizada">
                        <complexContent>
                            <extension base="ans:ct_diariaProrrogacao">
```

```
                                        <sequence>
                                                <element name="quantidadeAutorizada"
type="ans:st_quantidade" minOccurs="0"/>
                                                <element name="motivoGlosa"
type="ans:ct_motivoGlosa" minOccurs="0"/>
                                        </sequence>
                                </extension>
                        </complexContent>
                </complexType>
                <complexType name="ct_diariaProrrogacao">
                        <sequence>
                                <element name="tipoAcomodacao" type="ans:st_tipoAcomodacao"/>
                                <element name="quantidadeDiaria" type="ans:st_quantidade"/>
                        </sequence>
                </complexType>
                <complexType name="ct_endereco">
                        <sequence>
                                <element name="tipoLogradouro" type="ans:st_tipoLogradouro"/>
                                <element name="logradouro" type="ans:st_logradouro"/>
                                <element name="numero" type="ans:st_numeroLogradouro"/>
                                <element name="complemento" type="ans:st_descricao15"
nillable="true" minOccurs="0"/>
                                <element name="codigoIBGEMunicipio"
type="ans:st_codigoMunicipioIBGE"/>
                                <element name="municipio" type="ans:st_descricao40"
minOccurs="0"/>
                                <element name="codigoUF" type="ans:st_UF"/>
                                <element name="cep" type="ans:st_CEP"/>
                        </sequence>
                </complexType>
                <complexType name="ct_executanteComplementar">
                        <sequence>
                                <element name="codigoProfissionalCompl"
type="ans:ct_identificacaoPrestador"/>
                                <element name="nomeProfissionalCompl" type="ans:st_nome"/>
                        </sequence>
                </complexType>
                <complexType name="ct_hipoteseDiagnostica">
                        <sequence>
                                <element name="CIDPrincipal" type="ans:ct_diagnosticoGuia"/>
                                <element name="diagnosticosSecundarios"
type="ans:ct_diagnosticosSecundarios" nillable="true" minOccurs="0"/>
                        </sequence>
                </complexType>
                <complexType name="ct_identificacaoEquipe">
                        <sequence>
                                <element name="membroEquipe" maxOccurs="unbounded">
                                        <complexType>
                                                <sequence>
                                                        <element name="codigoProfissional"
type="ans:ct_identificacaoPrestadorEquipe"/>
                                                        <element name="identificacaoProfissional"
type="ans:ct_profissionalExecutante"/>
                                                        <element name="cpf" type="ans:st_CPF"
minOccurs="0"/>
                                                        <element name="posicaoProfissional"
type="ans:st_posicaoProfissao"/>
                                                </sequence>
                                        </complexType>
                                </element>
```

```
            </sequence>
        </complexType>
        <complexType name="ct_identificacaoOperadora">
            <sequence>
                <element name="registroANS" type="ans:st_registroANS"/>
                <element name="nomeOperadora" type="ans:st_nome"/>
                <element name="CNPJ" type="ans:st_CNPJ"/>
            </sequence>
        </complexType>
        <complexType name="ct_identificacaoPrestador">
            <choice>
                <element name="CNPJ" type="ans:st_CNPJ"/>
                <element name="cpf" type="ans:st_CPF"/>
                <element name="codigoPrestadorNaOperadora"
type="ans:st_codigoPrestadorNaOperadora"/>
                <element name="conselhoProfissional"
type="ans:ct_conselhoProfissional"/>
            </choice>
        </complexType>
        <complexType name="ct_identificacaoPrestadorEquipe">
            <choice>
                <element name="cpf" type="ans:st_CPF"/>
                <element name="codigoPrestadorNaOperadora"
type="ans:st_codigoPrestadorNaOperadora"/>
                <element name="conselhoProfissional"
type="ans:ct_conselhoProfissional"/>
            </choice>
        </complexType>
        <complexType name="ct_identificacaoPrestadorExecutante">
            <choice>
                <element name="CNPJ" type="ans:st_CNPJ"/>
                <element name="CPF" type="ans:st_CPF"/>
                <element name="codigoPrestadorNaOperadora"
type="ans:st_codigoPrestadorNaOperadora"/>
            </choice>
        </complexType>
        <complexType name="ct_identificacaoProfissional">
            <sequence>
                <element name="nomeProfissional" type="ans:st_nome"
minOccurs="0"/>
                <element name="conselhoProfissional"
type="ans:ct_conselhoProfissional"/>
                <element name="cbos" type="ans:st_CBOS" minOccurs="0"/>
            </sequence>
        </complexType>
        <complexType name="ct_identificacaoProfissionalCompl">
            <choice>
                <element name="cpf" type="ans:st_CPF"/>
                <element name="codigoPrestadorNaOperadora"
type="ans:st_codigoPrestadorNaOperadora"/>
            </choice>
        </complexType>
        <complexType name="ct_identificadorPrestOdontoDemo">
            <choice>
                <element name="CNPJ" type="ans:st_CNPJ"/>
                <element name="cpf" type="ans:st_CPF"/>
            </choice>
        </complexType>
        <complexType name="ct_internacaoObstetrica">
            <sequence>
```

```
                        <element name="emGestacao" type="ans:st_simNao"
minOccurs="0"/>
                        <element name="aborto" type="ans:st_simNao" minOccurs="0"/>
                        <element name="transtornoMaternoRelGravidez"
type="ans:st_simNao" minOccurs="0"/>
                        <element name="complicacaoPeriodoPuerperio"
type="ans:st_simNao" minOccurs="0"/>
                        <element name="atendimentoRNSalaParto" type="ans:st_simNao"
minOccurs="0"/>
                        <element name="complicacaoNeonatal" type="ans:st_simNao"
minOccurs="0"/>
                        <element name="baixoPeso" type="ans:st_simNao" minOccurs="0"/>
                        <element name="partoCesareo" type="ans:st_simNao"
minOccurs="0"/>
                        <element name="partoNormal" type="ans:st_simNao"
minOccurs="0"/>
                        <element name="declaracoesNascidosVivos"
type="ans:ct_numeroDN" minOccurs="0"/>
                        <element name="qtdNascidosVivosTermo"
type="ans:st_numeroInteiro" minOccurs="0"/>
                        <element name="qtdNascidosMortos" type="ans:st_numeroInteiro"
minOccurs="0"/>
                        <element name="qtdVivosPrematuros" type="ans:st_numeroInteiro"
minOccurs="0"/>
                        <element name="obitoMulher" type="ans:st_obitoMulher"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_itemSolicitacao">
                <sequence>
                        <element name="identificacaoProcedimentos" type="ans:ct_tabela"/>
                        <element name="quantidadeSolicitada" type="ans:st_quantidade"/>
                        <element name="quantidadeAutorizada" type="ans:st_quantidade"
minOccurs="0"/>
                        <element name="statusSolicitacaoProcedimento"
type="ans:st_statusSolicitacao"/>
                        <element name="glosas" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element name="motivoGlosa"
type="ans:ct_motivoGlosa" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_loteOdonto">
                <sequence>
                        <element name="numeroLote" type="ans:st_descricao12"/>
                        <element name="detalheGuia" type="ans:ct_detalheGuiaOdonto"/>
                        <element name="valorTotalLiberadoLote"
type="ans:st_valorMonetario"/>
                        <element name="valorTotalProcessadoLote"
type="ans:st_valorMonetario"/>
                        <element name="valorTotalGlosaLote" type="ans:st_valorMonetario"/>
                </sequence>
        </complexType>
        <complexType name="ct_motivoGlosa">
```

```
                <sequence>
                        <element name="codigoGlosa" type="ans:st_tipoGlosa"/>
                        <element name="descricaoGlosa" type="ans:st_descricaoGlosa"/>
                </sequence>
        </complexType>
        <complexType name="ct_numeroDN">
                <sequence>
                        <element name="numeroDN" type="ans:st_descricao15"
minOccurs="0" maxOccurs="10"/>
                </sequence>
        </complexType>
        <complexType name="ct_obitoInternacao">
                <sequence>
                        <element name="CID" type="ans:ct_diagnostico"/>
                        <element name="numeroDeclaracao" type="ans:st_descricao12"/>
                </sequence>
        </complexType>
        <complexType name="ct_obitoNeonatal">
                <choice>
                        <element name="qtdeobitoPrecoce" type="ans:st_quantidadeObito"
minOccurs="0"/>
                        <element name="qtdeobitoTardio" type="ans:st_quantidadeObito"
minOccurs="0"/>
                </choice>
        </complexType>
        <complexType name="ct_OPMSolicitadas">
                <sequence>
                        <element name="OPMSolicitada" maxOccurs="unbounded">
                                <complexType>
                                        <sequence>
                                                <element name="quantidadeSolicitada"
type="ans:st_quantidade"/>
                                                <element name="fabricante"
type="ans:st_descricao40" minOccurs="0"/>
                                                <element name="valor"
type="ans:st_valorMonetario" minOccurs="0"/>
                                                <element name="OPM" type="ans:ct_tabela"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_OPMUtilizada">
                <sequence>
                        <element name="OPM">
                                <complexType>
                                        <sequence>
                                                <element name="identificacaoOPM"
maxOccurs="unbounded">
                                                        <complexType>
                                                                <sequence>
                                                                        <element
name="OPM" type="ans:ct_tabela"/>
                                                                        <element
name="quantidade" type="ans:st_quantidade"/>
                                                                        <element
name="codigoBarra" type="ans:st_descricao20" minOccurs="0"/>
                                                                        <element
name="valorUnitario" type="ans:st_valorMonetario" minOccurs="0"/>
```

```
                                                                    <element
name="valorTotal" type="ans:st_valorMonetario" minOccurs="0"/>
                                                                </sequence>
                                                            </complexType>
                                                        </element>
                                                    </sequence>
                                                </complexType>
                                            </element>
                                            <element name="valorTotalOPM" type="ans:st_valorMonetario"
minOccurs="0"/>
                                        </sequence>
                                    </complexType>
                                    <complexType name="ct_OPMUtilizadasRevisao">
                                        <sequence>
                                            <element name="OPMUtilizada">
                                                <complexType>
                                                    <sequence>
                                                        <element name="OPM"
maxOccurs="unbounded">
                                                            <complexType>
                                                                <sequence>
                                                                    <element
name="OPM" type="ans:ct_tabela"/>
                                                                    <element
name="valorUnitario" type="ans:st_valorMonetario" minOccurs="0"/>
                                                                    <element
name="quantidade" type="ans:st_quantidade"/>
                                                                    <element
name="codigoBarra" type="ans:st_descricao20" minOccurs="0"/>
                                                                    <element
name="valorTotal" type="ans:st_valorMonetario" minOccurs="0"/>
                                                                    <element
name="justificativa" type="ans:st_justificativa"/>
                                                                </sequence>
                                                            </complexType>
                                                        </element>
                                                    </sequence>
                                                </complexType>
                                            </element>
                                            <element name="valorTotalOPM" type="ans:st_valorMonetario"
minOccurs="0"/>
                                        </sequence>
                                    </complexType>
                                    <complexType name="ct_outrasDespesas">
                                        <sequence>
                                            <element name="despesa" maxOccurs="unbounded">
                                                <complexType>
                                                    <sequence>
                                                        <element name="identificadorDespesa"
type="ans:ct_tabela"/>
                                                        <element name="tipoDespesa"
type="ans:st_outrasDespesas"/>
                                                        <element name="dataRealizacao"
type="ans:st_data"/>
                                                        <element name="horaInicial"
type="ans:st_hora" minOccurs="0"/>
                                                        <element name="horaFinal"
type="ans:st_hora" minOccurs="0"/>
                                                        <element name="reducaoAcrescimo"
type="ans:st_percentual" minOccurs="0"/>
```

```
                                        <element name="quantidade"
type="ans:st_quantidade"/>
                                        <element name="valorUnitario"
type="ans:st_valorMonetario" minOccurs="0"/>
                                        <element name="valorTotal"
type="ans:st_valorMonetario" minOccurs="0"/>
                                    </sequence>
                                </complexType>
                        </element>
                        <element name="totalGeralOutrasDespesas"
type="ans:st_valorMonetario" minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_outrasDespesasRevisao">
                <sequence>
                        <element name="itensDespesas" maxOccurs="unbounded">
                                <complexType>
                                        <sequence>
                                                <element name="tipoDespesa"
type="ans:st_outrasDespesas"/>
                                                <element name="dataRealizacao"
type="ans:st_data"/>
                                                <element name="despesa"
type="ans:ct_tabela"/>
                                                <element name="reducaoAcrescimo"
type="ans:st_percentual" minOccurs="0"/>
                                                <element name="quantidade"
type="ans:st_quantidade"/>
                                                <element name="valorUnitario"
type="ans:st_valorMonetario" minOccurs="0"/>
                                                <element name="valorTotal"
type="ans:st_valorMonetario" minOccurs="0"/>
                                                <element name="justificativa"
type="ans:st_justificativa"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element name="totalGeralOutrasDespesas"
type="ans:st_valorMonetario" minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_periodoProcessamento">
                <sequence>
                        <element name="dataInicial" type="ans:st_data"/>
                        <element name="dataFinal" type="ans:st_data"/>
                </sequence>
        </complexType>
        <complexType name="ct_procedimentoOdontologia">
                <sequence>
                        <element name="procedimentoOdonto" type="ans:ct_tabela"/>
                        <element name="denteRegiao" type="ans:ct_denteRegiao"
minOccurs="0"/>
                        <element name="face" type="ans:st_face" minOccurs="0"/>
                        <element name="quantidade" type="ans:st_quantidadeProcedimento"
minOccurs="0"/>
                        <element name="quantidadeUS" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="valor" type="ans:st_valorMonetario" minOccurs="0"/>
                        <element name="franquiaCoparticipacao"
type="ans:st_valorMonetario" minOccurs="0"/>
```

```
            </sequence>
        </complexType>
        <complexType name="ct_procedimentoOdontoRealizado">
            <sequence>
                <element name="procedimentoOdonto" type="ans:ct_tabela"/>
                <element name="denteRegiao" type="ans:ct_denteRegiao"
minOccurs="0"/>
                <element name="face" type="ans:st_face" minOccurs="0"/>
                <element name="dataRealizacao" type="ans:st_data"/>
                <element name="quantidade" type="ans:st_quantidadeProcedimento"
minOccurs="0"/>
                <element name="valorProcessado" type="ans:st_valorMonetario"/>
                <element name="valorGlosaEstorno" type="ans:st_valorMonetario"
minOccurs="0"/>
                <element name="motivoGlosa" type="ans:st_tipoGlosa"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="valorLiberado" type="ans:st_valorMonetario"/>
            </sequence>
        </complexType>
        <complexType name="ct_procedimentosRealizados">
            <sequence>
                <element name="procedimentoRealizado" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <element name="data" type="ans:st_data"/>
                            <element name="horaInicio"
type="ans:st_hora" minOccurs="0"/>
                            <element name="horaFim" type="ans:st_hora"
minOccurs="0"/>
                            <element name="procedimento"
type="ans:ct_tabela"/>
                            <element name="quantidadeRealizada"
type="ans:st_quantidadeProcedimento"/>
                            <element name="viaAcesso"
type="ans:st_viaDeAcesso" minOccurs="0"/>
                            <element name="tecnicaUtilizada"
type="ans:st_tecnicaUtilizada" minOccurs="0"/>
                            <element name="reducaoAcrescimo"
type="ans:st_percentual" minOccurs="0"/>
                            <element name="valor"
type="ans:st_valorMonetario" nillable="true" minOccurs="0"/>
                            <element name="valorTotal"
type="ans:st_valorMonetario" nillable="true" minOccurs="0"/>
                        </sequence>
                    </complexType>
                </element>
            </sequence>
        </complexType>
        <complexType name="ct_procedimentoOdontoRevisao">
            <sequence>
                <element name="procedimentoOdontologia">
                    <complexType>
                        <complexContent>
                            <extension
base="ans:ct_procedimentoOdontologia">
                                <sequence>
                                    <element
name="dataRealizacao" type="ans:st_data"/>
                                </sequence>
                            </extension>
```

```
                                        </complexContent>
                                    </complexType>
                                </element>
                                <element name="valorTotal" type="ans:st_valorMonetario"
minOccurs="0"/>
                                <element name="quantidadeTotalUS" type="ans:st_valorMonetario"
minOccurs="0"/>
                                <element name="totalFranquiaCoparticipacao"
type="ans:st_valorMonetario" minOccurs="0"/>
                                <element name="justificativaRevisao" type="ans:st_justificativa"/>
                        </sequence>
                </complexType>
                <complexType name="ct_procedimentoRealizadoEquipe">
                        <sequence>
                                <element name="procedimentos" maxOccurs="unbounded">
                                        <complexType>
                                                <sequence>
                                                        <element name="equipe"
type="ans:ct_identificacaoEquipe" minOccurs="0"/>
                                                        <element name="procedimento"
type="ans:ct_tabela"/>
                                                        <element name="data" type="ans:st_data"/>
                                                        <element name="horaInicio"
type="ans:st_hora" minOccurs="0"/>
                                                        <element name="horaFim" type="ans:st_hora"
minOccurs="0"/>
                                                        <element name="quantidadeRealizada"
type="ans:st_quantidadeProcedimento"/>
                                                        <element name="viaAcesso"
type="ans:st_viaDeAcesso" minOccurs="0"/>
                                                        <element name="tecnicaUtilizada"
type="ans:st_tecnicaUtilizada" minOccurs="0"/>
                                                        <element name="reducaoAcrescimo"
type="ans:st_percentual" minOccurs="0"/>
                                                        <element name="valor"
type="ans:st_valorMonetario" nillable="true" minOccurs="0"/>
                                                        <element name="valorTotal"
type="ans:st_valorMonetario" nillable="true" minOccurs="0"/>
                                                </sequence>
                                        </complexType>
                                </element>
                        </sequence>
                </complexType>
                <complexType name="ct_procedimentoRevisao">
                        <sequence>
                                <element name="procedimentosRevisao" maxOccurs="unbounded">
                                        <complexType>
                                                <sequence>
                                                        <element name="data" type="ans:st_data"/>
                                                        <element name="horaInicio"
type="ans:st_hora" minOccurs="0"/>
                                                        <element name="horaFim" type="ans:st_hora"
minOccurs="0"/>
                                                        <element name="procedimento"
type="ans:ct_tabela"/>
                                                        <element name="quantidadeRealizada"
type="ans:st_quantidadeProcedimento"/>
                                                        <element name="viaAcesso"
type="ans:st_viaDeAcesso" minOccurs="0"/>
```

```
                                        <element name="tecnicaUtilizada"
type="ans:st_tecnicaUtilizada" minOccurs="0"/>
                                        <element name="reducaoAcrescimo"
type="ans:st_percentual" minOccurs="0"/>
                                        <element name="valor"
type="ans:st_valorMonetario" nillable="true" minOccurs="0"/>
                                        <element name="valorTotal"
type="ans:st_valorMonetario" nillable="true" minOccurs="0"/>
                                        <element name="justificativaRevisao"
type="ans:st_justificativa" minOccurs="0"/>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
                </complexType>
                <complexType name="ct_procedimentosSolicitados">
                        <sequence>
                            <element name="procedimentoSolicitado" maxOccurs="unbounded">
                                <complexType>
                                    <sequence>
                                        <element name="quantidadeSolicitada"
type="ans:st_quantidadeProcedimento"/>
                                        <element name="procedimento"
type="ans:ct_tabela"/>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
                </complexType>
                <complexType name="ct_profissionalExecutante">
                        <sequence>
                            <element name="nomeExecutante" type="ans:st_nome"
minOccurs="0"/>
                            <element name="conselhoProfissional"
type="ans:ct_conselhoProfissional"/>
                            <element name="codigoCBOS" type="ans:st_CBOS" minOccurs="0"/>
                        </sequence>
                </complexType>
                <complexType name="ct_protocoloRecebimento">
                        <sequence>
                            <element name="dadosOperadora"
type="ans:ct_identificacaoOperadora"/>
                            <element name="dadosPrestador" type="ans:ct_contratado"/>
                            <element name="dataEnvioLote" type="ans:st_data"/>
                            <element name="numeroLote" type="ans:st_descricao12"/>
                            <element name="mensagemDetalheProtocolo"
type="ans:ct_detalheProtocolo" minOccurs="0"/>
                            <choice>
                                <element name="numeroProtocoloRecebimento"
type="ans:st_protocolo"/>
                                <element name="mensagemErro" type="ans:st_observacao"/>
                            </choice>
                        </sequence>
                </complexType>
                <complexType name="ct_reciboCancelaGuia">
                        <sequence>
                            <element name="operadora" type="ans:ct_identificacaoOperadora"/>
                            <element name="guiasCanceladas">
                                <complexType>
                                    <sequence>
```

```
                                        <element name="dadosGuia"
maxOccurs="unbounded">
                                            <complexType>
                                                <complexContent>
                                                    <extension
base="ans:ct_cabecalhoGuia">
                                                        <sequence>

        <element name="statusCancelamento" type="ans:st_statusCancelamento"/>

        <element name="observacao" type="ans:st_observacao" minOccurs="0"/>
                                                        </sequence>
                                                    </extension>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
        </complexType>
        <complexType name="ct_retornoElegibilidade">
                <sequence>
                    <element name="dadosOperadora"
type="ans:ct_identificacaoOperadora"/>
                    <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                    <element name="respostaSolicitacao" type="ans:st_simNao"/>
                    <element name="motivoNegacao" type="ans:ct_motivoGlosa"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_solicitaStatusAutorizacao">
                <sequence>
                    <element name="identificacaoAutorizacao">
                        <complexType>
                            <sequence>
                                <element name="identificacaoSolicitacao"
maxOccurs="unbounded">
                                    <complexType>
                                        <complexContent>
                                            <extension
base="ans:ct_cabecalhoSolicitacao">
                                                <sequence>

        <element name="profissionalSolicitante" type="ans:ct_contratadoSolicitante"/>

        <element name="identificacaoBeneficiario" type="ans:ct_beneficiario" minOccurs="0"/>
                                                </sequence>
                                            </extension>
                                        </complexContent>
                                    </complexType>
                                </element>
                            </sequence>
                        </complexType>
                    </element>
                    <element name="dataSolicitaStatus" type="ans:st_dataHora"/>
                </sequence>
        </complexType>
        <complexType name="ct_situacaoAutorizacao">
                <sequence>
```

```
                    <element name="dadosAutorizacao" maxOccurs="unbounded">
                        <complexType>
                            <complexContent>
                                <extension
base="ans:ct_autorizacaoProcedimento">
                                    <sequence>
                                        <element name="totalOdonto"
minOccurs="0">
                                            <complexType>
                                                <sequence>

        <element name="totalvalor" type="ans:st_quantidade" minOccurs="0"/>

        <element name="totalFranquiaCoparticipacao" type="ans:st_quantidade"
minOccurs="0"/>

        <element name="totalquantidadeUS" type="ans:st_quantidade" minOccurs="0"/>
                                                </sequence>
                                            </complexType>
                                        </element>
                                        <element
name="ct_diariaProrrogacaoAutorizada" minOccurs="0">
                                            <complexType>

        <complexContent>

        <extension base="ans:ct_diariaProrrogacao">

        <sequence>

                <element name="tipoAcomodacaoAutorizada" type="ans:st_tipoAcomodacao"
minOccurs="0"/>

                <element name="quantidadeAutorizada" type="ans:st_quantidade"
minOccurs="0"/>

                <element name="motivoGlosa" type="ans:ct_motivoGlosa" minOccurs="0"/>

        </sequence>

        </extension>

        </complexContent>
                                            </complexType>
                                        </element>
                                    </sequence>
                                </extension>
                            </complexContent>
                        </complexType>
                    </element>
                </sequence>
        </complexType>
        <complexType name="ct_situacaoClinica">
                <sequence>
                    <element name="dentes" maxOccurs="unbounded">
                        <complexType>
                            <sequence>
                                <element name="elementoDentario"
type="ans:st_dente"/>
```

```
                                        <element name="condicaoClinica"
type="ans:st_condicaoClinica"/>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
                    </complexType>
                    <complexType name="ct_situacaoDentes">
                        <sequence>
                            <element name="dente" type="ans:st_dente"/>
                            <element name="obturado" type="ans:st_simNao"/>
                            <element name="careado" type="ans:st_simNao"/>
                            <element name="perdido" type="ans:st_simNao"/>
                        </sequence>
                    </complexType>
                    <complexType name="ct_situacaoInicial">
                        <sequence>
                            <element name="situacaoClinica" type="ans:ct_situacaoClinica"/>
                            <element name="doencaPeriodontal" type="ans:st_simNao"
minOccurs="0"/>
                            <element name="alteracaoTecidoMole" type="ans:st_simNao"
minOccurs="0"/>
                            <element name="observacao" type="ans:st_observacao"
minOccurs="0"/>
                        </sequence>
                    </complexType>
                    <complexType name="ct_solicitaElegibilidade">
                        <sequence>
                            <element name="dadosPrestador" type="ans:ct_contratado"/>
                            <element name="dadosBeneficiario" type="ans:ct_beneficiario"/>
                        </sequence>
                    </complexType>
                    <complexType name="ct_solicitante">
                        <sequence>
                            <element name="contratado" type="ans:ct_contratadoSolicitante"/>
                            <element name="profissional"
type="ans:ct_identificacaoProfissional"/>
                        </sequence>
                    </complexType>
                    <complexType name="ct_solicitacaoDemonstrativoPagamento">
                        <sequence>
                            <element name="dadosPrestador" type="ans:ct_contratado"/>
                            <element name="registroANS" type="ans:st_registroANS"/>
                            <element name="dataSolicitacao" type="ans:st_data"/>
                            <element name="tipoDemonstrativo"
type="ans:st_tipoDemonstrativo"/>
                            <choice>
                                <sequence>
                                    <element name="dataInicial" type="ans:st_data"/>
                                    <element name="dataFinal" type="ans:st_data"
minOccurs="0"/>
                                </sequence>
                                <element name="numeroProtocolo" type="ans:st_protocolo"/>
                            </choice>
                        </sequence>
                    </complexType>
                    <complexType name="ct_solicitacaoStatusProtocolo">
                        <sequence>
                            <element name="registroANS" type="ans:st_registroANS"/>
                            <element name="dadosPrestador" type="ans:ct_contratado"/>
```

```
                        <element name="dataSolicitacao" type="ans:st_data"/>
                        <element name="detalheProtocolo">
                                <complexType>
                                        <sequence>
                                                <element name="numeroProtocolo"
type="ans:st_protocolo" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_statusLote">
                <sequence>
                        <element name="numeroProtocolo" type="ans:st_protocolo"/>
                        <element name="numeroLote" type="ans:st_inteiro12"/>
                        <element name="dataEnvioLote" type="ans:st_data"/>
                        <element name="numeroFatura" type="ans:st_inteiro12"
minOccurs="0"/>
                        <element name="valorProcessado" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="valorLiberado" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="valorGlosa" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="status" type="ans:st_statusProtocolo"/>
                        <element name="guias">
                                <complexType>
                                        <sequence>
                                                <element name="detalheguia" minOccurs="0"
maxOccurs="unbounded">
                                                        <complexType>
                                                                <sequence>
                                                                        <element
name="identificacaoGuia" type="ans:ct_cabecalhoGuia"/>
                                                                        <element
name="procedimentosAnalisados">

        <complexType>

        <sequence>

        <element name="procedimento" type="ans:ct_itemSolicitacao"
maxOccurs="unbounded"/>

        </sequence>

        </complexType>
                                                                        </element>
                                                                </sequence>
                                                        </complexType>
                                                </element>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_StatusProtocolo">
                <sequence>
                        <element name="registroANS" type="ans:st_registroANS"/>
                        <element name="nomeOperadora" type="ans:st_nome"/>
```

```
                        <element name="CNPJ" type="ans:st_CNPJ"/>
                        <element name="statusProtocolo" type="ans:st_statusProtocolo"
minOccurs="0"/>
                        <element name="dadosPrestador" type="ans:ct_contratado"/>
                        <element name="CNESPrestador" type="ans:st_CNES"
minOccurs="0"/>
                        <element name="lote">
                                <complexType>
                                        <sequence>
                                                <element name="detalheLote"
type="ans:ct_statusLote" maxOccurs="unbounded"/>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ct_tabela">
                <sequence>
                        <element name="codigo" type="ans:st_codigoTabela"/>
                        <element name="tipoTabela" type="ans:st_tabela"/>
                        <element name="descricao" type="ans:st_descricaoTabela"
nillable="true" minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_tempoEvolucaoDoenca">
                <sequence>
                        <element name="valor" type="ans:st_numeroInteiro"/>
                        <element name="unidadeTempo" type="ans:st_unidadeTempo"/>
                </sequence>
        </complexType>
        <complexType name="ct_totalDemoOdonto">
                <sequence>
                        <element name="valorTotalGeralGlosa"
type="ans:st_valorMonetario"/>
                        <element name="valorTotalGeralLiberado"
type="ans:st_valorMonetario"/>
                        <element name="valorTotalProcessadoGuia"
type="ans:st_valorMonetario"/>
                </sequence>
        </complexType>
        <complexType name="ct_totaisOdonto">
                <sequence>
                        <element name="servicoExecutado" type="ans:st_valorMonetario"/>
                        <element name="taxas" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="materiais" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="medicamentos" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="totalGeral" type="ans:st_valorMonetario"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ct_tratamenteOdontoRealizado">
                <sequence>
                        <element name="data" type="ans:st_data"/>
                        <element name="procedimento" type="ans:ct_tabela"/>
                        <element name="dente" type="ans:st_dente"/>
                        <element name="face" type="ans:st_face"/>
                </sequence>
```

```
        </complexType>
        <complexType name="ct_tratamentoOdontoSolicitado">
                <sequence>
                        <element name="procedimento" type="ans:ct_tabela"/>
                        <element name="dente" type="ans:st_dente"/>
                        <element name="face" type="ans:st_face"/>
                </sequence>
        </complexType>
        <complexType name="ct_valorTotalServicos">
                <sequence>
                        <element name="servicosExecutados" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="diarias" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="taxas" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="materiais" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="medicamentos" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="gases" type="ans:st_valorMonetario"
minOccurs="0"/>
                        <element name="totalGeral" type="ans:st_valorMonetario"
minOccurs="0"/>
                </sequence>
        </complexType>
</schema>
```

Nome do arquivo: tissSimpleTypesV2_01_03.xsd

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ans="http://www.ans.gov.br/padroes/tiss/schemas"
targetNamespace="http://www.ans.gov.br/padroes/tiss/schemas"
elementFormDefault="qualified">
        <!-- Schema com os tipos simples de dados re-utilizados na construção dos tipos
complexos -->
        <simpleType name="st_agencia">
                <restriction base="string">
                        <maxLength value="10"/>
                </restriction>
        </simpleType>
        <simpleType name="st_atendimentoOdonto">
                <restriction base="string">
                        <length value="1"/>
                        <enumeration value="1"/>
                        <enumeration value="2"/>
                        <enumeration value="3"/>
                        <enumeration value="4"/>
                        <!--Tratamento Odontológico-->
                        <!--Exame Radiológico-->
                        <!--Ortodontia-->
                        <!--Urgência/Emergência-->
                </restriction>
        </simpleType>
        <simpleType name="st_banco">
                <restriction base="integer">
                        <totalDigits value="10"/>
                </restriction>
        </simpleType>
```

```
<simpleType name="st_CBOS">
        <restriction base="string">
                <enumeration value="01280"/>
                <enumeration value="03370"/>
                <enumeration value="04945"/>
                <enumeration value="05110"/>
                <enumeration value="05230"/>
                <enumeration value="05250"/>
                <enumeration value="05270"/>
                <enumeration value="05290"/>
                <enumeration value="06105"/>
                <enumeration value="06110"/>
                <enumeration value="06112"/>
                <enumeration value="06113"/>
                <enumeration value="06114"/>
                <enumeration value="06115"/>
                <enumeration value="06116"/>
                <enumeration value="06117"/>
                <enumeration value="06118"/>
                <enumeration value="06119"/>
                <enumeration value="06120"/>
                <enumeration value="06121"/>
                <enumeration value="06122"/>
                <enumeration value="06123"/>
                <enumeration value="06124"/>
                <enumeration value="06125"/>
                <enumeration value="06126"/>
                <enumeration value="06127"/>
                <enumeration value="06128"/>
                <enumeration value="06129"/>
                <enumeration value="06130"/>
                <enumeration value="06131"/>
                <enumeration value="06132"/>
                <enumeration value="06133"/>
                <enumeration value="06134"/>
                <enumeration value="06135"/>
                <enumeration value="06136"/>
                <enumeration value="06137"/>
                <enumeration value="06138"/>
                <enumeration value="06139"/>
                <enumeration value="06140"/>
                <enumeration value="06141"/>
                <enumeration value="06142"/>
                <enumeration value="06143"/>
                <enumeration value="06144"/>
                <enumeration value="06145"/>
                <enumeration value="06146"/>
                <enumeration value="06147"/>
                <enumeration value="06148"/>
                <enumeration value="06149"/>
                <enumeration value="06150"/>
                <enumeration value="06151"/>
                <enumeration value="06152"/>
                <enumeration value="06153"/>
                <enumeration value="06154"/>
                <enumeration value="06155"/>
                <enumeration value="06156"/>
                <enumeration value="06157"/>
                <enumeration value="06158"/>
                <enumeration value="06159"/>
```

```
<enumeration value="06160"/>
<enumeration value="06161"/>
<enumeration value="06162"/>
<enumeration value="06163"/>
<enumeration value="06164"/>
<enumeration value="06165"/>
<enumeration value="06166"/>
<enumeration value="06167"/>
<enumeration value="06168"/>
<enumeration value="06170"/>
<enumeration value="06172"/>
<enumeration value="06175"/>
<enumeration value="06177"/>
<enumeration value="06180"/>
<enumeration value="06190"/>
<enumeration value="06310"/>
<enumeration value="06330"/>
<enumeration value="06335"/>
<enumeration value="06340"/>
<enumeration value="06345"/>
<enumeration value="06350"/>
<enumeration value="06355"/>
<enumeration value="06360"/>
<enumeration value="06365"/>
<enumeration value="06370"/>
<enumeration value="06510"/>
<enumeration value="06540"/>
<enumeration value="06710"/>
<enumeration value="06810"/>
<enumeration value="07110"/>
<enumeration value="07111"/>
<enumeration value="07112"/>
<enumeration value="07130"/>
<enumeration value="07140"/>
<enumeration value="07145"/>
<enumeration value="07150"/>
<enumeration value="07155"/>
<enumeration value="07160"/>
<enumeration value="07165"/>
<enumeration value="07210"/>
<enumeration value="07215"/>
<enumeration value="07220"/>
<enumeration value="07230"/>
<enumeration value="07230"/>
<enumeration value="07231"/>
<enumeration value="07234"/>
<enumeration value="07236"/>
<enumeration value="07238"/>
<enumeration value="07290"/>
<enumeration value="07310"/>
<enumeration value="07410"/>
<enumeration value="07525"/>
<enumeration value="07530"/>
<enumeration value="07550"/>
<enumeration value="07620"/>
<enumeration value="07630"/>
<enumeration value="07690"/>
<enumeration value="07720"/>
<enumeration value="07730"/>
<enumeration value="07740"/>
```

```
<enumeration value="07790"/>
<enumeration value="07914"/>
<enumeration value="07915"/>
<enumeration value="07925"/>
<enumeration value="07935"/>
<enumeration value="07945"/>
<enumeration value="07946"/>
<enumeration value="07948"/>
<enumeration value="07950"/>
<enumeration value="19998"/>
<enumeration value="19999"/>
<enumeration value="31120"/>
<enumeration value="31120"/>
<enumeration value="31122"/>
<enumeration value="57210"/>
<enumeration value="57216"/>
<enumeration value="57220"/>
<enumeration value="57232"/>
<enumeration value="57233"/>
<enumeration value="57235"/>
<enumeration value="57260"/>
<enumeration value="57275"/>
<enumeration value="57280"/>
<enumeration value="57282"/>
<enumeration value="99999"/>
<!-- Físico nuclear-->
<!--Técnico de saneamento-->
<!-- Pedagogo-->
<!-- Biologista em geral (biólogo biomédico) -->
<!-- Bioquímico-->
<!-- Bacteriologista-->
<!-- Farmacologista-->
<!-- Outros bacteriologistas e trabalhadores assemelhados-->
<!-- Médico em geral (clinico geral) -->
<!-- Médico cirurgião em geral-->
<!-- Médico cirurgião pediátrico-->
<!-- Médico de perícias médicas-->
<!-- Médico anatomopatologista-->
<!-- Médico anestesista-->
<!--Médico broncoesofalogista-->
<!--Médico cardiologista-->
<!--Médico cirurgião vascular-->
<!-- Médico dermatologista-->
<!-- Médico cirurgião cardiovascular-->
<!-- Médico oncologista pediátrico-->
<!-- Médico do trabalho-->
<!-- Médico gastroenterologista-->
<!-- Médico hematologista-->
<!-- Médico endocrinologista-->
<!-- Médico nuclear-->
<!-- Médico endoscopista-->
<!-- Médico fisiatra-->
<!-- Médico oncologista clinico-->
<!-- Médico reumatologista-->
<!-- Médico neurocirurgião-->
<!-- Médico ginecologista-->
<!-- Médico alergista/imunologista-->
<!-- Médico geriatra-->
<!-- Médico hemoterapeuta-->
<!-- Médico hansenologista-->
```

<!-- Médico legista-->
<!-- Médico nefrologista-->
<!-- Médico mastologista-->
<!-- Médico sanitarista-->
<!-- Médico de saúde da família-->
<!-- Médico neurologista-->
<!-- Médico geneticista clinico-->
<!-- Médico infectologista-->
<!-- Médico obstetra-->
<!-- Médico cirurgião de cabeça e pescoço-->
<!-- Médico oftalmologista-->
<!-- Médico homeopata-->
<!-- Médico ginecologista / obstetra-->
<!-- Médico ortopedista-->
<!-- Médico geral comunitário-->
<!-- Médico otorrinolaringologista-->
<!-- Médico citopatologista-->
<!-- Médico cirurgião torácico-->
<!-- Médico pediatra-->
<!-- Médico ultrasonografista-->
<!--Médico pneumotisiologista-->
<!--Médico acupunturista-->
<!-- Médico cancerologista-->
<!-- Médico proctologista-->
<!-- Médico cirurgião de mão-->
<!-- Médico psiquiatra-->
<!-- Médico cirurgião do aparelho digestivo-->
<!-- Médico plantonista-->
<!-- Médico radiologista-->
<!-- Médico intensivista-->
<!-- Médico radioterapeuta-->
<!-- Médico oncologista cirúrgico-->
<!-- Médico urologista-->
<!-- Médico patologista clinico-->
<!-- Médico angiologista-->
<!-- Médico de medicina esportiva-->
<!-- Médico cirurgião plástico-->
<!-- Outros médicos-->
<!-- Cirurgião dentista em geral-->
<!-- Cirurgião dentista (saúde pública) -->
<!-- Cirurgião dentista (traumatologia buco maxilo facial) -->
<!-- Cirurgião dentista (endodontia) -->
<!-- Cirurgião dentista (ortodontia) -->
<!-- Cirurgião dentista (patologia bucal) -->
<!-- Cirurgião dentista (pediatria) -->
<!-- Cirurgião dentista (prótese) -->
<!-- Cirurgião dentista (radiologia) -->
<!-- Cirurgião dentista (periodontia)-->
<!-- Médico veterinário em geral-->
<!-- Zootecnista-->
<!-- Farmacêutico em geral-->
<!-- Nutricionista em geral-->
<!-- Enfermeiro, em geral-->
<!-- Enfermeiro do PACS-->
<!-- Enfermeiro do PSF-->
<!-- Enfermeiro sanitarista-->
<!-- Enfermeiro do trabalho-->
<!-- Enfermeiro obstetra-->
<!-- Enfermeiro de centro cirúrgico-->
<!-- Enfermeiro de terapia intensiva-->

```
                        <!-- Enfermeiro peuricultor e pediátrico-->
                        <!-- Enfermeiro psiquiátrico-->
                        <!-- Técnico de enfermagem em geral-->
                        <!-- Técnico de enfermagem do trabalho-->
                        <!-- Técnico de enfermagem de terapia intensiva-->
                        <!-- Técnico de enfermagem psiquiátrica-->
                        <!-- Fiscal sanitário-->
                        <!-- Técnico em vigilância sanitária-->
                        <!-- Técnico em reabilitação-->
                        <!-- Técnico em equipamento médico-hospitalar-->
                        <!-- Técnico de enfermagem do PSF-->
                        <!-- Assistente social em geral-->
                        <!-- Psicólogo em geral-->
                        <!-- Ortoptista-->
                        <!-- Ótico-->
                        <!-- Técnico em orientação e mobilidade de cegos e deficientes
visuais-->
                        <!-- Fisioterapeuta-->
                        <!-- Terapeuta ocupacional-->
                        <!-- Terapeutas em geral (outros terapeutas) -->
                        <!-- Operador de raio-x-->
                        <!-- Operador de eletrocardiógrafo-->
                        <!-- Operador de eletroencefalografo-->
                        <!-- Outros operadores de equipamentos médicos e odontológicos-->
                        <!-- Foniatra-->
                        <!-- Acupunturista-->
                        <!-- Fonoaudiólogo-->
                        <!-- Técnico de higiene dental-->
                        <!-- Quiropata-->
                        <!-- Técnico em laboratório-->
                        <!-- Técnico de radiologia-->
                        <!-- Técnico de ortopedia-->
                        <!-- Outros profissionais de nível superior-->
                        <!-- Outros profissionais de nível técnico e médio-->
                        <!-- Agente Administrativo-->
                        <!-- Técnico administrativo-->
                        <!-- Auxiliar administrativo-->
                        <!-- Auxiliar de enfermagem-->
                        <!-- Auxiliar de enfermagem no PSF-->
                        <!-- Atendente de enfermagem-->
                        <!-- Agente de vigilância sanitária-->
                        <!-- Agente de saneamento-->
                        <!-- Guarda de endemias/Agente de zoonoses/Agente de controle de
vetores-->
                        <!-- Parteira-->
                        <!-- Auxiliar de análises clínicas-->
                        <!-- Socorrista habilitado-->
                        <!-- Agente comunitário-->
                        <!-- Outros profissionais não classificáveis nessa tabela-->
                </restriction>
        </simpleType>
        <simpleType name="st_CEP">
                <restriction base="string">
                        <maxLength value="8"/>
                </restriction>
        </simpleType>
        <simpleType name="st_CNES">
                <restriction base="string">
                        <maxLength value="7"/>
                </restriction>
```

```
</simpleType>
<simpleType name="st_CNPJ">
        <restriction base="string">
                <pattern value="[0-9]{14}"/>
        </restriction>
</simpleType>
<simpleType name="st_CPF">
        <restriction base="string">
                <pattern value="[0-9]{11}"/>
        </restriction>
</simpleType>
<simpleType name="st_codigoBeneficiario">
        <restriction base="string">
                <maxLength value="20"/>
                <minLength value="17"/>
        </restriction>
</simpleType>
<simpleType name="st_codigoDiagnostico">
        <restriction base="string">
                <maxLength value="5"/>
        </restriction>
</simpleType>
<simpleType name="st_codigoMunicipioIBGE">
        <restriction base="string">
                <length value="7"/>
        </restriction>
</simpleType>
<simpleType name="st_codigoPrestadorNaOperadora">
        <restriction base="string">
                <maxLength value="20"/>
        </restriction>
</simpleType>
<simpleType name="st_codigoTabela">
        <restriction base="string">
                <maxLength value="10"/>
                <minLength value="8"/>
        </restriction>
</simpleType>
<simpleType name="st_condicaoClinica">
        <restriction base="string">
                <maxLength value="1"/>
                <enumeration value="H"/>
                <enumeration value="E"/>
                <enumeration value="A"/>
                <enumeration value="C"/>
                <enumeration value="R"/>
                <!--Hígido-->
                <!--Extração indicada-->
                <!--Ausente-->
                <!--Cariado-->
                <!--Restaurado-->
        </restriction>
</simpleType>
<simpleType name="st_conta">
        <restriction base="string">
                <maxLength value="20"/>
        </restriction>
</simpleType>
<simpleType name="st_conselhoProfissional">
        <restriction base="string">
```

```
                <enumeration value="CRAS"/>
                <enumeration value="COREN"/>
                <enumeration value="CRF"/>
                <enumeration value="CREFITO"/>
                <enumeration value="CRM"/>
                <enumeration value="CRV"/>
                <enumeration value="CRN"/>
                <enumeration value="CRO"/>
                <enumeration value="CRP"/>
                <enumeration value="OUT"/>
                <enumeration value="CRFA"/>
                <!--Conselho Regional de Assistência Social-->
                <!--Conselho Federal de Enfermagem-->
                <!--Conselho Regional de Farmácia-->
                <!--Conselho Regional de Fisioterapia e Terapia Ocupacional-->
                <!--Conselho Regional de Medicina-->
                <!--Conselho Regional de Medicina Veterinária-->
                <!--Conselho Regional de Nutrição-->
                <!--Conselho Regional de Odontologia-->
                <!--Conselho Regional de Psicologia-->
                <!--Outros Conselhos-->
                <!--Conselho Regional de Fonoaudiologia-->
        </restriction>
</simpleType>
<simpleType name="st_data">
        <restriction base="string">
                <pattern value="[0-9]{2}/[0-9]{2}/[0-9]{4}"/>
        </restriction>
</simpleType>
<simpleType name="st_dataHora">
        <restriction base="string">
                <pattern value="[0-9]{2}/[0-9]{2}/[0-9]{4}H[0-9]{2}:[0-9]{2}"/>
        </restriction>
</simpleType>
<simpleType name="st_dente">
        <restriction base="string">
                <maxLength value="2"/>
                <enumeration value="11"/>
                <enumeration value="12"/>
                <enumeration value="13"/>
                <enumeration value="14"/>
                <enumeration value="15"/>
                <enumeration value="16"/>
                <enumeration value="17"/>
                <enumeration value="18"/>
                <enumeration value="19"/>
                <enumeration value="21"/>
                <enumeration value="22"/>
                <enumeration value="23"/>
                <enumeration value="24"/>
                <enumeration value="25"/>
                <enumeration value="26"/>
                <enumeration value="27"/>
                <enumeration value="28"/>
                <enumeration value="29"/>
                <enumeration value="31"/>
                <enumeration value="32"/>
                <enumeration value="33"/>
                <enumeration value="34"/>
                <enumeration value="35"/>
```

```
                    <enumeration value="36"/>
                    <enumeration value="37"/>
                    <enumeration value="38"/>
                    <enumeration value="39"/>
                    <enumeration value="41"/>
                    <enumeration value="42"/>
                    <enumeration value="43"/>
                    <enumeration value="44"/>
                    <enumeration value="45"/>
                    <enumeration value="46"/>
                    <enumeration value="47"/>
                    <enumeration value="48"/>
                    <enumeration value="49"/>
                    <enumeration value="51"/>
                    <enumeration value="52"/>
                    <enumeration value="53"/>
                    <enumeration value="54"/>
                    <enumeration value="55"/>
                    <enumeration value="59"/>
                    <enumeration value="61"/>
                    <enumeration value="62"/>
                    <enumeration value="63"/>
                    <enumeration value="64"/>
                    <enumeration value="65"/>
                    <enumeration value="69"/>
                    <enumeration value="71"/>
                    <enumeration value="72"/>
                    <enumeration value="73"/>
                    <enumeration value="74"/>
                    <enumeration value="75"/>
                    <enumeration value="79"/>
                    <enumeration value="81"/>
                    <enumeration value="82"/>
                    <enumeration value="83"/>
                    <enumeration value="84"/>
                    <enumeration value="85"/>
                    <enumeration value="89"/>
                    <!--Incisivo Central Superior Direito-->
                    <!--Incisivo Lateral Superior Direito-->
                    <!--Canino Superior Direito-->
                    <!--Primeiro Pré-molar Superior Direito-->
                    <!--Segundo Pré-molar Superior Direito-->
                    <!--Primeiro Molar Superior Direito-->
                    <!--Segundo Molar Superior Direito-->
                    <!--Terceiro Molar Superior Direito-->
                    <!-- Dente Permanente Supranumerário em Hemi-arco Superior Direito
-->
                    <!--Incisivo Central Superior Esquerdo-->
                    <!--Incisivo Lateral Superior Esquerdo-->
                    <!--Canino Superior Esquerdo-->
                    <!--Primeiro Pré-molar Superior Esquerdo-->
                    <!--Segundo Pré-molar Superior Esquerdo-->
                    <!--Primeiro Molar Superior Esquerdo-->
                    <!--Segundo Molar Superior Esquerdo-->
                    <!--Terceiro Molar Superior Esquerdo-->
                    <!--Dente Permanente Supranumerário em Hemi-arco Superior
Esquerdo-->
                    <!--Incisivo Central Inferior Esquerdo-->
                    <!--Incisivo Lateral Inferior Esquerdo-->
                    <!--Canino Inferior Esquerdo-->
```

```
                        <!--Primeiro Pré-molar Inferior Esquerdo-->
                        <!--Segundo Pré-molar Inferior Esquerdo-->
                        <!--Primeiro Molar Inferior Esquerdo-->
                        <!--Segundo Molar Inferior Esquerdo-->
                        <!--Terceiro Molar Inferior Esquerdo-->
                        <!--Dente Permanente Supranumerário em Hemi-arco Inferior
Esquerdo-->
                        <!--Incisivo Central Inferior Direito-->
                        <!--Incisivo Lateral Inferior Direito-->
                        <!--Canino Inferior Direito-->
                        <!--Primeiro Pré-molar Inferior Direito-->
                        <!--Segundo Pré-molar Inferior Direito-->
                        <!--Primeiro Molar Inferior Direito-->
                        <!--Segundo Molar Inferior Direito-->
                        <!--Terceiro Molar Inferior Direito-->
                        <!--Dente Permanente Supranumerário em Hemi-arco Inferior Direito--
>
                        <!--Incisivo Central Decíduo Superior Direito-->
                        <!--Incisivo Lateral Decíduo Superior Direito-->
                        <!--Canino Decíduo Superior Direito-->
                        <!--Primeiro Molar Decíduo Superior Direito-->
                        <!--Segundo Molar Decíduo Superior Direito-->
                        <!--Dente Decíduo Supranumerário em Hemi-arco Superior Direito-->
                        <!--Incisivo Central Decíduo Superior Esquerdo-->
                        <!--Incisivo Lateral Decíduo Superior Esquerdo-->
                        <!--Canino Decíduo Superior Esquerdo-->
                        <!--Primeiro Molar Decíduo Superior Esquerdo-->
                        <!--Segundo Molar Decíduo Superior Esquerdo-->
                        <!--Dente Decíduo Supranumerário em Hemi-arco Superior Esquerdo--
>
                        <!--Incisivo Central Decíduo Inferior Esquerdo-->
                        <!--Incisivo Lateral Decíduo Inferior Esquerdo-->
                        <!--Canino Decíduo Inferior Esquerdo-->
                        <!--Primeiro Molar Decíduo Inferior Esquerdo-->
                        <!--Segundo Molar Decíduo Inferior Esquerdo-->
                        <!--Dente Decíduo Supranumerário em Hemi-arco Inferior Esquerdo-->
                        <!--Incisivo Central Decíduo Inferior Direito-->
                        <!--Incisivo Lateral Decíduo Inferior Direito-->
                        <!--Canino Decíduo Inferior Direito-->
                        <!--Primeiro Molar Decíduo Inferior Direito-->
                        <!--Segundo molar Decíduo Inferior Direito-->
                        <!--Dente Decíduo Supranumerário em Hemi-arco Inferior Direito-->
                </restriction>
        </simpleType>
        <simpleType name="st_descricao12">
                <restriction base="string">
                        <maxLength value="12"/>
                </restriction>
        </simpleType>
        <simpleType name="st_descricao15">
                <restriction base="string">
                        <maxLength value="15"/>
                </restriction>
        </simpleType>
        <simpleType name="st_descricao20">
                <restriction base="string">
                        <maxLength value="20"/>
                </restriction>
        </simpleType>
        <simpleType name="st_descricao40">
```

```
        <restriction base="string">
            <maxLength value="40"/>
        </restriction>
</simpleType>
<simpleType name="st_descricaoGlosa">
        <restriction base="string">
            <maxLength value="100"/>
        </restriction>
</simpleType>
<simpleType name="st_descricaoTabela">
        <restriction base="string">
            <maxLength value="60"/>
        </restriction>
</simpleType>
<simpleType name="st_diariasUTI">
        <restriction base="integer">
            <totalDigits value="3"/>
        </restriction>
</simpleType>
<simpleType name="st_eletivaEmergencia">
        <restriction base="string">
            <enumeration value="E"/>
            <enumeration value="U"/>
            <!--Eletiva-->
            <!--Urgência-->
        </restriction>
</simpleType>
<simpleType name="st_face">
        <restriction base="string">
            <maxLength value="5"/>
        </restriction>
</simpleType>
<simpleType name="st_formaPagamento">
        <restriction base="string">
            <enumeration value="1"/>
            <enumeration value="2"/>
            <enumeration value="3"/>
            <!-- Crédito em conta -->
            <!-- Carteira -->
            <!-- Boleto bancário -->
        </restriction>
</simpleType>
<simpleType name="st_hora">
        <restriction base="string">
            <pattern value="[0-9]{2}:[0-9]{2}"/>
        </restriction>
</simpleType>
<simpleType name="st_indicacaoClinica">
        <restriction base="string">
            <maxLength value="500"/>
        </restriction>
</simpleType>
<simpleType name="st_indicadorAcidente">
        <restriction base="string">
            <enumeration value="0"/>
            <enumeration value="1"/>
            <enumeration value="2"/>
            <!--Acidente ou doença relacionada a trabalho -->
            <!--Acidente ou doença relacionada a trânsito -->
            <!--Outros -->
```

```
        </restriction>
</simpleType>
<simpleType name="st_inteiro12">
        <restriction base="integer">
                <totalDigits value="12"/>
        </restriction>
</simpleType>
<simpleType name="st_justificativa">
        <restriction base="string">
                <maxLength value="500"/>
        </restriction>
</simpleType>
<simpleType name="st_logradouro">
        <restriction base="string">
                <maxLength value="40"/>
        </restriction>
</simpleType>
<simpleType name="st_motivoSaida">
        <restriction base="string">
                <enumeration value="10"/>
                <enumeration value="11"/>
                <enumeration value="12"/>
                <enumeration value="13"/>
                <enumeration value="14"/>
                <enumeration value="15"/>
                <enumeration value="16"/>
                <enumeration value="17"/>
                <enumeration value="18"/>
                <enumeration value="19"/>
                <enumeration value="21"/>
                <enumeration value="22"/>
                <enumeration value="23"/>
                <enumeration value="24"/>
                <enumeration value="25"/>
                <enumeration value="31"/>
                <enumeration value="32"/>
                <enumeration value="33"/>
                <enumeration value="34"/>
                <enumeration value="35"/>
                <enumeration value="36"/>
                <enumeration value="37"/>
                <enumeration value="38"/>
                <enumeration value="39"/>
                <enumeration value="41"/>
                <enumeration value="43"/>
                <enumeration value="44"/>
                <enumeration value="45"/>
                <enumeration value="46"/>
                <enumeration value="51"/>
                <enumeration value="52"/>
                <enumeration value="53"/>
                <enumeration value="54"/>
                <enumeration value="61"/>
                <enumeration value="62"/>
                <enumeration value="63"/>
                <enumeration value="64"/>
                <enumeration value="65"/>
                <enumeration value="66"/>
                <enumeration value="67"/>
                <enumeration value="68"/>
```

```
                        <enumeration value="69"/>
                        <enumeration value="71"/>
                        <!--Para complementação em internação domiciliar-->
                        <!--Curado-->
                        <!--Melhorado-->
                        <!--Inalterado-->
                        <!--A pedido-->
                        <!--Internado para diagnóstico-->
                        <!--Administrativa-->
                        <!--Por indisciplina-->
                        <!--Por evasão-->
                        <!--Para completar tratamento-->
                        <!--Por caracteristicas da doença-->
                        <!--Por intercorrência-->
                        <!--Por motivo social-->
                        <!--Por doença crônica-->
                        <!--Por impossibilidade de convivio sócio-familiar-->
                        <!--Para tisiologia-->
                        <!--Para psiquiatria-->
                        <!--Para clinica médica-->
                        <!--Para cirurgia-->
                        <!--Para obstetrícia-->
                        <!--Para berçário-->
                        <!--Para pediatria-->
                        <!--Para isolamento-->
                        <!--Para outros (CTI, radioterapia, etc) -->
                        <!--Até 24 horas da internação-->
                        <!--Ocorreu após 48 horas da internação-->
                        <!--Óbito de parturiente, com necrópsia, com permanência do recém-
nascido-->
                        <!--De 24 ate 48h de internação, paciente chegou agônico-->
                        <!--De 24 ate 48h de internação, paciente não chegou agônico-->
                        <!--Nas primeiras 48 horas, chegou agônico-->
                        <!--Nas primeiras 48 horas, não chegou agônico-->
                        <!--Ocorreu após 48 horas de internação-->
                        <!--Óbito de parturiente, com necrópsia, com permanência do recém-
nascido-->
                        <!--Em politraumatizado c/menos 24hs da 1ª Cirurgia-->
                        <!--Em politraumatizado 24 a 48hs após 1ª Cirurgia-->
                        <!--Em politraumatizado 48 a 72hs após 1ª Cirurgia-->
                        <!--Em politraumatizado acima 72hs após 1ª Cirurgia-->
                        <!--Em cirurgia de emergência c/menos 24hs após 1ª Cirurgia-->
                        <!--Em cirurgia de emergência 24 a 48hs após 1ª Cirurgia-->
                        <!--Em cirurgia de emergência 48 a 72hs após 1ª Cirurgia-->
                        <!--Em cirurgia emergência acima 72h após 1ª cirurgia-->
                        <!--Alta da parturiente com permanência do recém-nascido-->
                        <!--Em caso de alta da parturiente com permanência do recém
nascido-->
                </restriction>
        </simpleType>
        <simpleType name="st_nome">
                <restriction base="string">
                        <maxLength value="70"/>
                </restriction>
        </simpleType>
        <simpleType name="st_numeroGuia">
                <restriction base="string">
                        <maxLength value="20"/>
                </restriction>
        </simpleType>
```

```
<simpleType name="st_numeroInteiro">
        <restriction base="integer"/>
</simpleType>
<simpleType name="st_numeroLogradouro">
        <restriction base="string">
                <maxLength value="5"/>
        </restriction>
</simpleType>
<simpleType name="st_numeroTel">
        <restriction base="string">
                <maxLength value="10"/>
        </restriction>
</simpleType>
<simpleType name="st_obito">
        <restriction base="string">
                <enumeration value="F"/>
                <enumeration value="N"/>
                <!--Fetal-->
                <!--Não Fetal-->
        </restriction>
</simpleType>
<simpleType name="st_obitoMulher">
        <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <!--Grávida no momento-->
                <!--Até 42 dias após o termino da gestação-->
                <!--De 43 dias até 12 meses após o término da gestação          -->
        </restriction>
</simpleType>
<simpleType name="st_observacao">
        <restriction base="string">
                <maxLength value="240"/>
        </restriction>
</simpleType>
<simpleType name="st_outrasDespesas">
        <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <enumeration value="4"/>
                <enumeration value="5"/>
                <enumeration value="6"/>
                <!-- Gases Medicinais -->
                <!-- Medicamentos -->
                <!-- Materiais -->
                <!-- Taxas Diversas -->
                <!-- Diárias -->
                <!-- Aluguel -->
        </restriction>
</simpleType>
<simpleType name="st_percentual">
        <restriction base="decimal">
                <totalDigits value="5"/>
                <fractionDigits value="2"/>
        </restriction>
</simpleType>
<simpleType name="st_posicaoProfissao">
        <restriction base="string">
```

```
                    <enumeration value="00"/>
                    <enumeration value="01"/>
                    <enumeration value="02"/>
                    <enumeration value="03"/>
                    <enumeration value="04"/>
                    <enumeration value="05"/>
                    <enumeration value="06"/>
                    <enumeration value="07"/>
                    <enumeration value="08"/>
                    <enumeration value="09"/>
                    <enumeration value="10"/>
                    <enumeration value="11"/>
                    <enumeration value="12"/>
                    <enumeration value="13"/>
                    <!--Cirurgião-->
                    <!--Primeiro Auxiliar-->
                    <!--Segundo Auxiliar-->
                    <!--Terceiro Auxiliar-->
                    <!--Quarto Auxiliar-->
                    <!--Instrumentador-->
                    <!--Anestesista-->
                    <!--Auxiliar de Anestesista-->
                    <!--Consultor-->
                    <!--Perfusionista-->
                    <!--Pediatra na sala de parto-->
                    <!--Auxiliar SADT-->
                    <!--Clínico-->
                    <!--Intensivista-->
            </restriction>
    </simpleType>
    <simpleType name="st_procedimentoSUS">
            <restriction base="string"/>
    </simpleType>
    <simpleType name="st_protocolo">
            <restriction base="string">
                    <maxLength value="12"/>
            </restriction>
    </simpleType>
    <simpleType name="st_quantidade">
            <restriction base="decimal">
                    <totalDigits value="12"/>
                    <fractionDigits value="4"/>
            </restriction>
    </simpleType>
    <simpleType name="st_quantidadeProcedimento">
            <restriction base="integer">
                    <totalDigits value="3"/>
            </restriction>
    </simpleType>
    <simpleType name="st_quantidadeObito">
            <restriction base="integer">
                    <totalDigits value="2"/>
            </restriction>
    </simpleType>
    <simpleType name="st_regiao">
            <restriction base="string">
                    <maxLength value="5"/>
                    <enumeration value="AI"/>
                    <enumeration value="AS"/>
                    <enumeration value="HASD"/>
```

```
<enumeration value="HASE"/>
<enumeration value="HAID"/>
<enumeration value="HAIE"/>
<enumeration value="ASAI"/>
<enumeration value="S1"/>
<enumeration value="S2"/>
<enumeration value="S3"/>
<enumeration value="S4"/>
<enumeration value="S5"/>
<enumeration value="S6"/>
<enumeration value="LG"/>
<enumeration value="CL"/>
<enumeration value="AB"/>
<enumeration value="PA"/>
<enumeration value="MJ"/>
<enumeration value="PD"/>
<enumeration value="PM"/>
<enumeration value="RM"/>
<enumeration value="MA"/>
<enumeration value="GI"/>
<enumeration value="PT"/>
<enumeration value="TP"/>
<enumeration value="RIS"/>
<enumeration value="RCSD"/>
<enumeration value="RCID"/>
<enumeration value="RMSD"/>
<enumeration value="RCSE"/>
<enumeration value="RPSE"/>
<enumeration value="RMSE"/>
<enumeration value="RII"/>
<enumeration value="RCID"/>
<enumeration value="RPID"/>
<enumeration value="RMID"/>
<enumeration value="RCIE"/>
<enumeration value="RPIE"/>
<enumeration value="RMIE"/>
<enumeration value="RMD"/>
<enumeration value="RME"/>
<enumeration value="RPD"/>
<enumeration value="RPE"/>
<enumeration value="RMPE"/>
<enumeration value="RMPD"/>
<enumeration value="SM"/>
<enumeration value="TU"/>
<enumeration value="SI"/>
<enumeration value="FLI"/>
<enumeration value="FLA"/>
<enumeration value="UV"/>
<enumeration value="PP"/>
<enumeration value="PI"/>
<enumeration value="LS"/>
<enumeration value="LI"/>
<enumeration value="RL"/>
<enumeration value="RP"/>
<enumeration value="RV"/>
<enumeration value="RSMD"/>
<enumeration value="RSME"/>
<enumeration value="RSL"/>
<!--Arco Inferior-->
<!--Arco Superior-->
```

```
        <!--Hemi-Arco Superior Direito-->
        <!--Hemi-Arco Superior Esquerdo-->
        <!--Hemi-Arco Inferior Direito-->
        <!--Hemi-Arco Inferior Esquerdo-->
        <!--Arcadas Superiores e inferiores-->
        <!--Sextante superior posterior direito-->
        <!--Sextante superior anterior-->
        <!--Sextante superior posterior esquerdo-->
        <!--Sextante inferior posterior esquerdo-->
        <!--Sextante inferior anterior-->
        <!--Sextante inferior posterior direito-->
        <!--Lingua-->
        <!--Comissura labial-->
        <!--Assoalho de boca-->
        <!--Palato-->
        <!--Mucosa jugal-->
        <!--Palato duro-->
        <!--Palato mole-->
        <!--Região retromolar-->
        <!--Mucosa alveolar-->
        <!--Gengiva inserida-->
        <!--Parótida-->
        <!--Tonsilas palatinas-->
        <!--Região dos Incisivos centrais superiores-->
        <!--Região do canino e lateral superior direito-->
        <!--Região dos pré-molares superiores direito-->
        <!--Região dos molares superiores direito-->
        <!--Região do canino e lateral superior esquerdo-->
        <!--Região dos pré-molares superiores esquerdo-->
        <!--Região dos molares superiores esquerdo-->
        <!--Região dos incisivos inferiores-->
        <!--Região de canino inferior direito-->
        <!--Região dos pré-molares inferiores direito-->
        <!--Região dos molares inferiores direito-->
        <!--Região de canino inferior esquerdo-->
        <!--Região dos pré-molares inferiores esquerdo-->
        <!--Região dos molares inferiores esquerdo-->
        <!--Região dos molares lado direito-->
        <!--Região dos molares lado esquerdo-->
        <!--Região dos pré-molares lado direito-->
        <!--Região dos pré-molares lado esquerdo-->
        <!--Região dos molares e pré-molares lado esquerdo-->
        <!--Região dos molares e pré-molares lado direito-->
        <!--Região do assoalho do seio maxilar-->
        <!--Região do Túber-->
        <!--Região de Sínfise-->
        <!--Freio lingual-->
        <!--Freios labiais-->
        <!--Úvula-->
        <!--Pregas palatinas-->
        <!--Papila incisiva-->
        <!--Lábio Superior-->
        <!--Lábio inferior-->
        <!--Região lingual-->
        <!--Região palatina-->
        <!--Região vestibular-->
        <!--Região Sub-Mandibular Direita-->
        <!--Região Sub-Mandibular Esquerda-->
        <!--Região Sub-Lingual-->
</restriction>
```

```
</simpleType>
<simpleType name="st_registroANS">
        <restriction base="string">
                <maxLength value="6"/>
        </restriction>
</simpleType>
<simpleType name="st_regimeInternacao">
        <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <!--Hospitalar-->
                <!--Hospital_dia-->
                <!--Domiciliar-->
        </restriction>
</simpleType>
<simpleType name="st_senhaAutorizacao">
        <restriction base="string">
                <maxLength value="20"/>
        </restriction>
</simpleType>
<simpleType name="st_simNao">
        <restriction base="boolean"/>
</simpleType>
<simpleType name="st_statusCancelamento">
        <restriction base="string">
                <maxLength value="1"/>
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <!--Cancelado com sucesso-->
                <!--Não cancelado-->
                <!--Guia inexistente-->
        </restriction>
</simpleType>
<simpleType name="st_statusProtocolo">
        <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <enumeration value="4"/>
                <enumeration value="5"/>
                <enumeration value="6"/>
                <enumeration value="7"/>
                <!--Recebido-->
                <!--Em análise-->
                <!--Liberado para pagamento-->
                <!--Encerrado sem pagamento-->
                <!--Analisado e aguardando liberação para o pagamento-->
                <!--Pagamento efetuado-->
                <!--Protocolo não localizado-->
        </restriction>
</simpleType>
<simpleType name="st_statusSolicitacao">
        <restriction base="string">
                <length value="1"/>
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <!-- Autorizado -->
```

```
                <!-- Em análise -->
                <!-- Negado -->
            </restriction>
        </simpleType>
        <simpleType name="st_tabela">
            <restriction base="string">
                <enumeration value="01"/>
                <enumeration value="02"/>
                <enumeration value="03"/>
                <enumeration value="04"/>
                <enumeration value="05"/>
                <enumeration value="06"/>
                <enumeration value="07"/>
                <enumeration value="08"/>
                <enumeration value="09"/>
                <enumeration value="10"/>
                <enumeration value="11"/>
                <enumeration value="12"/>
                <enumeration value="13"/>
                <enumeration value="14"/>
                <enumeration value="15"/>
                <enumeration value="94"/>
                <enumeration value="95"/>
                <enumeration value="96"/>
                <enumeration value="97"/>
                <enumeration value="98"/>
                <enumeration value="99"/>
                <enumeration value="00"/>
                <!--Tabela AMB-90-->
                <!--Tabela AMB 92-->
                <!--Tabela AMB 96-->
                <!--Tabela AMB 99-->
                <!--Tabela Brasindice-->
                <!--Classificação Brasileira Hierarquizada de Procedimentos Médicos--
>
                <!--Tabela CIEFAS 93-->
                <!-- Tabela CIEFAS 2000-->
                <!--Rol de procedimentos ANS-->
                <!--Tabela de procedimentos ambulatoriais SUS-->
                <!--Tabela de procedimentos hospitalares SUS-->
                <!--Tabela SIMPRO-->
                <!--Tabela TUNEP-->
                <!--Tabela VRPO-->
                <!--Tabela de Intercâmbio Sistema Uniodonto-->
                <!--Tabela própria procedimento-->
                <!--Tabela própria materiais-->
                <!--Tabela própria medicamentos-->
                <!--Tabela própria taxas hospitalares-->
                <!--Tabela própria pacotes-->
                <!--Tabela própria gases medicinais-->
                <!--Outras tabelas-->
            </restriction>
        </simpleType>
        <simpleType name="st_tabelasDiagnostico">
            <restriction base="string">
                <enumeration value="CID-10"/>
            </restriction>
        </simpleType>
        <simpleType name="st_tecnicaUtilizada">
            <restriction base="string">
```

```
                    <enumeration value="V"/>
                    <enumeration value="C"/>
                    <!--Videolaparoscopia-->
                    <!--Convencional-->
            </restriction>
    </simpleType>
    <simpleType name="st_tipoAcomodacao">
            <restriction base="string">
                    <enumeration value="1"/>
                    <enumeration value="2"/>
                    <enumeration value="3"/>
                    <enumeration value="4"/>
                    <enumeration value="5"/>
                    <enumeration value="6"/>
                    <enumeration value="7"/>
                    <enumeration value="8"/>
                    <enumeration value="11"/>
                    <enumeration value="12"/>
                    <enumeration value="13"/>
                    <enumeration value="14"/>
                    <enumeration value="15"/>
                    <enumeration value="21"/>
                    <enumeration value="22"/>
                    <enumeration value="23"/>
                    <enumeration value="31"/>
                    <enumeration value="32"/>
                    <enumeration value="33"/>
                    <enumeration value="34"/>
                    <enumeration value="35"/>
                    <enumeration value="41"/>
                    <enumeration value="42"/>
                    <enumeration value="43"/>
                    <enumeration value="51"/>
                    <enumeration value="52"/>
                    <enumeration value="53"/>
                    <enumeration value="54"/>
                    <enumeration value="55"/>
                    <enumeration value="61"/>
                    <!--Enfermaria-->
                    <!--Quarto particular-->
                    <!--UTI-->
                    <!--Enfermaria dois leitos-->
                    <!--One Day clinic-->
                    <!--Unidade intermediaria-->
                    <!--Apartamento-->
                    <!--Ambulatório-->
                    <!--Apartamento luxo-->
                    <!--Apartamento Simples-->
                    <!--Apartamento Standard-->
                    <!--Apartamento Suíte-->
                    <!--Apartamento com alojamento conjunto-->
                    <!--Berçário normal-->
                    <!--Berçário patológico / prematuro-->
                    <!--Berçário patológico com isolamento-->
                    <!--Enfermaria (3 leitos) -->
                    <!--Enfermaria (4 ou mais leitos) -->
                    <!--Enfermaria com alojamento conjunto-->
                    <!--Hospital Dia-->
                    <!--Isolamento-->
                    <!--Quarto Coletivo (2 leitos) -->
```

```
                <!--Quarto privativo-->
                <!--Quarto com alojamento conjunto-->
                <!--UTI Adulto-->
                <!--UTI Pediátrica-->
                <!--UTI Neo-Natal-->
                <!--TSI - Unidade de Terapia semi-Intensiva-->
                <!--Unidade coronariana-->
                <!--Outras diárias-->
        </restriction>
</simpleType>
<simpleType name="st_tipoAtendimento">
        <restriction base="string">
                <enumeration value="01"/>
                <enumeration value="02"/>
                <enumeration value="03"/>
                <enumeration value="04"/>
                <enumeration value="05"/>
                <enumeration value="06"/>
                <enumeration value="07"/>
                <enumeration value="08"/>
                <enumeration value="09"/>
                <enumeration value="10"/>
                <!--Remoção-->
                <!--Pequena Cirurgia-->
                <!--Terapias-->
                <!--Consulta-->
                <!--Exames-->
                <!--Atendimento Domiciliar-->
                <!--SADT Internado-->
                <!--Quimioterapia-->
                <!--Radioterapia-->
                <!--Terapia Renal Substitutiva (TRS) -->
        </restriction>
</simpleType>
<simpleType name="st_tipoConsulta">
        <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <!-- Primeira Consulta-->
                <!--Seguimento-->
                <!--Pré-natal-->
        </restriction>
</simpleType>
<simpleType name="st_tipoDemonstrativo">
        <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <!-- Demonstrativo de Pagamento -->
                <!-- Demonstrativo de análise da conta médica -->
                <!-- Demonstrativo de Pagamento - Odontologia -->
        </restriction>
</simpleType>
<simpleType name="st_tipoDoenca">
        <restriction base="string">
                <enumeration value="A"/>
                <enumeration value="C"/>
                <!--Aguda-->
                <!--Cronica-->
```

```
        </restriction>
</simpleType>
<simpleType name="st_tipoFaturamento">
        <restriction base="string">
                <enumeration value="T"/>
                <enumeration value="P"/>
                <!--Total-->
                <!--Parcial-->
        </restriction>
</simpleType>
<simpleType name="st_tipoGlosa">
        <restriction base="string">
                <maxLength value="4"/>
                <enumeration value="1001"/>
                <enumeration value="1002"/>
                <enumeration value="1003"/>
                <enumeration value="1004"/>
                <enumeration value="1005"/>
                <enumeration value="1006"/>
                <enumeration value="1007"/>
                <enumeration value="1008"/>
                <enumeration value="1009"/>
                <enumeration value="1010"/>
                <enumeration value="1011"/>
                <enumeration value="1012"/>
                <enumeration value="1013"/>
                <enumeration value="1014"/>
                <enumeration value="1015"/>
                <enumeration value="1016"/>
                <enumeration value="1017"/>
                <enumeration value="1018"/>
                <enumeration value="1019"/>
                <enumeration value="1020"/>
                <enumeration value="1021"/>
                <enumeration value="1022"/>
                <enumeration value="1023"/>
                <enumeration value="1024"/>
                <enumeration value="1099"/>
                <enumeration value="1101"/>
                <enumeration value="1102"/>
                <enumeration value="1103"/>
                <enumeration value="1104"/>
                <enumeration value="1199"/>
                <enumeration value="1201"/>
                <enumeration value="1202"/>
                <enumeration value="1203"/>
                <enumeration value="1204"/>
                <enumeration value="1205"/>
                <enumeration value="1206"/>
                <enumeration value="1207"/>
                <enumeration value="1208"/>
                <enumeration value="1209"/>
                <enumeration value="1210"/>
                <enumeration value="1211"/>
                <enumeration value="1212"/>
                <enumeration value="1213"/>
                <enumeration value="1214"/>
                <enumeration value="1215"/>
                <enumeration value="1216"/>
                <enumeration value="1217"/>
```

```
<enumeration value="1299"/>
<enumeration value="1301"/>
<enumeration value="1302"/>
<enumeration value="1303"/>
<enumeration value="1304"/>
<enumeration value="1305"/>
<enumeration value="1306"/>
<enumeration value="1307"/>
<enumeration value="1308"/>
<enumeration value="1309"/>
<enumeration value="1310"/>
<enumeration value="1311"/>
<enumeration value="1312"/>
<enumeration value="1313"/>
<enumeration value="1314"/>
<enumeration value="1315"/>
<enumeration value="1316"/>
<enumeration value="1317"/>
<enumeration value="1318"/>
<enumeration value="1319"/>
<enumeration value="1320"/>
<enumeration value="1321"/>
<enumeration value="1399"/>
<enumeration value="1401"/>
<enumeration value="1402"/>
<enumeration value="1403"/>
<enumeration value="1404"/>
<enumeration value="1405"/>
<enumeration value="1406"/>
<enumeration value="1407"/>
<enumeration value="1408"/>
<enumeration value="1409"/>
<enumeration value="1410"/>
<enumeration value="1411"/>
<enumeration value="1412"/>
<enumeration value="1413"/>
<enumeration value="1414"/>
<enumeration value="1415"/>
<enumeration value="1416"/>
<enumeration value="1417"/>
<enumeration value="1418"/>
<enumeration value="1419"/>
<enumeration value="1420"/>
<enumeration value="1421"/>
<enumeration value="1422"/>
<enumeration value="1423"/>
<enumeration value="1424"/>
<enumeration value="1425"/>
<enumeration value="1426"/>
<enumeration value="1427"/>
<enumeration value="1428"/>
<enumeration value="1429"/>
<enumeration value="1430"/>
<enumeration value="1431"/>
<enumeration value="1432"/>
<enumeration value="1433"/>
<enumeration value="1499"/>
<enumeration value="1501"/>
<enumeration value="1502"/>
<enumeration value="1503"/>
```

```
<enumeration value="1504"/>
<enumeration value="1505"/>
<enumeration value="1506"/>
<enumeration value="1507"/>
<enumeration value="1508"/>
<enumeration value="1509"/>
<enumeration value="1599"/>
<enumeration value="1601"/>
<enumeration value="1602"/>
<enumeration value="1603"/>
<enumeration value="1604"/>
<enumeration value="1605"/>
<enumeration value="1606"/>
<enumeration value="1607"/>
<enumeration value="1608"/>
<enumeration value="1609"/>
<enumeration value="1610"/>
<enumeration value="1611"/>
<enumeration value="1612"/>
<enumeration value="1613"/>
<enumeration value="1614"/>
<enumeration value="1615"/>
<enumeration value="1699"/>
<enumeration value="1701"/>
<enumeration value="1702"/>
<enumeration value="1703"/>
<enumeration value="1704"/>
<enumeration value="1705"/>
<enumeration value="1706"/>
<enumeration value="1707"/>
<enumeration value="1708"/>
<enumeration value="1709"/>
<enumeration value="1710"/>
<enumeration value="1711"/>
<enumeration value="1712"/>
<enumeration value="1713"/>
<enumeration value="1714"/>
<enumeration value="1715"/>
<enumeration value="1716"/>
<enumeration value="1799"/>
<enumeration value="1801"/>
<enumeration value="1802"/>
<enumeration value="1803"/>
<enumeration value="1804"/>
<enumeration value="1805"/>
<enumeration value="1806"/>
<enumeration value="1807"/>
<enumeration value="1808"/>
<enumeration value="1809"/>
<enumeration value="1810"/>
<enumeration value="1811"/>
<enumeration value="1812"/>
<enumeration value="1813"/>
<enumeration value="1814"/>
<enumeration value="1815"/>
<enumeration value="1816"/>
<enumeration value="1817"/>
<enumeration value="1818"/>
<enumeration value="1819"/>
<enumeration value="1820"/>
```

```
<enumeration value="1821"/>
<enumeration value="1822"/>
<enumeration value="1823"/>
<enumeration value="1824"/>
<enumeration value="1825"/>
<enumeration value="1899"/>
<enumeration value="1901"/>
<enumeration value="1902"/>
<enumeration value="1903"/>
<enumeration value="1904"/>
<enumeration value="1905"/>
<enumeration value="1906"/>
<enumeration value="1907"/>
<enumeration value="1908"/>
<enumeration value="1909"/>
<enumeration value="1910"/>
<enumeration value="1911"/>
<enumeration value="1912"/>
<enumeration value="1913"/>
<enumeration value="1914"/>
<enumeration value="1915"/>
<enumeration value="1916"/>
<enumeration value="1999"/>
<enumeration value="2001"/>
<enumeration value="2002"/>
<enumeration value="2003"/>
<enumeration value="2004"/>
<enumeration value="2005"/>
<enumeration value="2006"/>
<enumeration value="2007"/>
<enumeration value="2008"/>
<enumeration value="2009"/>
<enumeration value="2010"/>
<enumeration value="2011"/>
<enumeration value="2012"/>
<enumeration value="2013"/>
<enumeration value="2014"/>
<enumeration value="2099"/>
<enumeration value="2101"/>
<enumeration value="2102"/>
<enumeration value="2103"/>
<enumeration value="2104"/>
<enumeration value="2105"/>
<enumeration value="2106"/>
<enumeration value="2107"/>
<enumeration value="2108"/>
<enumeration value="2109"/>
<enumeration value="2110"/>
<enumeration value="2111"/>
<enumeration value="2112"/>
<enumeration value="2113"/>
<enumeration value="2114"/>
<enumeration value="2199"/>
<enumeration value="2201"/>
<enumeration value="2202"/>
<enumeration value="2203"/>
<enumeration value="2204"/>
<enumeration value="2205"/>
<enumeration value="2206"/>
<enumeration value="2207"/>
```

<enumeration value="2208"/>
<enumeration value="2209"/>
<enumeration value="2210"/>
<enumeration value="2211"/>
<enumeration value="2299"/>
<enumeration value="2301"/>
<enumeration value="2302"/>
<enumeration value="2303"/>
<enumeration value="2304"/>
<enumeration value="2305"/>
<enumeration value="2306"/>
<enumeration value="2307"/>
<enumeration value="2308"/>
<enumeration value="2309"/>
<enumeration value="2310"/>
<enumeration value="2399"/>
<enumeration value="2401"/>
<enumeration value="2402"/>
<enumeration value="2403"/>
<enumeration value="2404"/>
<enumeration value="2405"/>
<enumeration value="2406"/>
<enumeration value="2407"/>
<enumeration value="2408"/>
<enumeration value="2409"/>
<enumeration value="2410"/>
<enumeration value="2411"/>
<enumeration value="2412"/>
<enumeration value="2413"/>
<enumeration value="2414"/>
<enumeration value="2415"/>
<enumeration value="2416"/>
<enumeration value="2417"/>
<enumeration value="2418"/>
<enumeration value="2419"/>
<enumeration value="2420"/>
<enumeration value="2421"/>
<enumeration value="2422"/>
<enumeration value="2423"/>
<enumeration value="2424"/>
<enumeration value="2499"/>
<enumeration value="2501"/>
<enumeration value="2502"/>
<enumeration value="2503"/>
<enumeration value="2504"/>
<enumeration value="2505"/>
<enumeration value="2506"/>
<enumeration value="2507"/>
<enumeration value="2508"/>
<enumeration value="2509"/>
<enumeration value="2510"/>
<enumeration value="2511"/>
<enumeration value="2512"/>
<enumeration value="2513"/>
<enumeration value="2514"/>
<enumeration value="2515"/>
<enumeration value="2516"/>
<enumeration value="2599"/>
<enumeration value="2601"/>
<enumeration value="2602"/>

<enumeration value="2603"/>
<enumeration value="2604"/>
<enumeration value="2605"/>
<enumeration value="2606"/>
<enumeration value="2607"/>
<enumeration value="2608"/>
<enumeration value="2609"/>
<enumeration value="2699"/>
<enumeration value="2701"/>
<enumeration value="2702"/>
<enumeration value="2703"/>
<enumeration value="2704"/>
<enumeration value="2705"/>
<enumeration value="2706"/>
<enumeration value="2707"/>
<enumeration value="2708"/>
<enumeration value="2709"/>
<enumeration value="2710"/>
<enumeration value="2711"/>
<enumeration value="2712"/>
<enumeration value="2713"/>
<enumeration value="2714"/>
<enumeration value="2715"/>
<enumeration value="2716"/>
<enumeration value="2717"/>
<enumeration value="2718"/>
<enumeration value="2799"/>
<enumeration value="2801"/>
<enumeration value="2802"/>
<enumeration value="2803"/>
<enumeration value="2804"/>
<enumeration value="2805"/>
<enumeration value="2806"/>
<enumeration value="2807"/>
<enumeration value="2808"/>
<enumeration value="2809"/>
<enumeration value="2810"/>
<enumeration value="2811"/>
<enumeration value="2812"/>
<enumeration value="2813"/>
<enumeration value="2814"/>
<enumeration value="2815"/>
<enumeration value="2816"/>
<enumeration value="2817"/>
<enumeration value="2818"/>
<enumeration value="2819"/>
<enumeration value="2820"/>
<enumeration value="2821"/>
<enumeration value="2822"/>
<enumeration value="2899"/>
<enumeration value="2901"/>
<enumeration value="2902"/>
<enumeration value="2903"/>
<enumeration value="2904"/>
<enumeration value="2905"/>
<enumeration value="2906"/>
<enumeration value="2999"/>
<enumeration value="3001"/>
<enumeration value="3002"/>
<enumeration value="3003"/>

```
<enumeration value="3004"/>
<enumeration value="3005"/>
<enumeration value="3006"/>
<enumeration value="3007"/>
<enumeration value="3008"/>
<enumeration value="3009"/>
<enumeration value="3010"/>
<enumeration value="3011"/>
<enumeration value="3012"/>
<enumeration value="3013"/>
<enumeration value="3014"/>
<enumeration value="3015"/>
<enumeration value="3016"/>
<enumeration value="3017"/>
<enumeration value="3018"/>
<enumeration value="3019"/>
<enumeration value="3020"/>
<enumeration value="3021"/>
<enumeration value="3022"/>
<enumeration value="3023"/>
<enumeration value="3024"/>
<enumeration value="3025"/>
<enumeration value="3026"/>
<enumeration value="3027"/>
<enumeration value="3028"/>
<enumeration value="3029"/>
<enumeration value="3030"/>
<enumeration value="3031"/>
<enumeration value="3032"/>
<enumeration value="3033"/>
<enumeration value="3034"/>
<enumeration value="3035"/>
<enumeration value="3036"/>
<enumeration value="3037"/>
<enumeration value="3038"/>
<enumeration value="3039"/>
<enumeration value="3040"/>
<enumeration value="3099"/>
<enumeration value="5001"/>
<enumeration value="5002"/>
<enumeration value="5003"/>
<enumeration value="5004"/>
<enumeration value="5005"/>
<enumeration value="5006"/>
<enumeration value="5007"/>
<enumeration value="5008"/>
<enumeration value="5009"/>
<enumeration value="5010"/>
<enumeration value="5011"/>
<enumeration value="5012"/>
<enumeration value="5013"/>
<enumeration value="5014"/>
<enumeration value="5015"/>
<enumeration value="5099"/>
<enumeration value="9901"/>
<enumeration value="9902"/>
<enumeration value="9903"/>
<enumeration value="9904"/>
<enumeration value="9905"/>
<enumeration value="9906"/>
```

```
<enumeration value="9907"/>
<enumeration value="9908"/>
<enumeration value="9909"/>
<enumeration value="9910"/>
<enumeration value="9911"/>
<enumeration value="9912"/>
<enumeration value="9913"/>
<enumeration value="9914"/>
<enumeration value="9915"/>
<enumeration value="9916"/>
<enumeration value="9917"/>
<enumeration value="9918"/>
<enumeration value="9919"/>
<enumeration value="9920"/>
<enumeration value="9921"/>
<enumeration value="9922"/>
<enumeration value="9923"/>
<enumeration value="9924"/>
<enumeration value="9925"/>
<enumeration value="9926"/>
<enumeration value="9927"/>
<enumeration value="9928"/>
<enumeration value="9929"/>
<enumeration value="9930"/>
<enumeration value="9931"/>
<enumeration value="9932"/>
<enumeration value="9933"/>
<enumeration value="9934"/>
<enumeration value="9935"/>
<enumeration value="9936"/>
<enumeration value="9937"/>
<enumeration value="9938"/>
<enumeration value="9939"/>
<enumeration value="9940"/>
<enumeration value="9941"/>
<enumeration value="9942"/>
<enumeration value="9943"/>
<enumeration value="9944"/>
<enumeration value="9945"/>
<enumeration value="9946"/>
<enumeration value="9947"/>
<enumeration value="9948"/>
<enumeration value="9949"/>
<enumeration value="9950"/>
<enumeration value="9951"/>
<enumeration value="9952"/>
<enumeration value="9953"/>
<enumeration value="9954"/>
<enumeration value="9955"/>
<enumeration value="9956"/>
<enumeration value="9957"/>
<enumeration value="9958"/>
<enumeration value="9959"/>
<enumeration value="9960"/>
<enumeration value="9961"/>
<enumeration value="9962"/>
<enumeration value="9963"/>
<enumeration value="9964"/>
<enumeration value="9965"/>
<enumeration value="9966"/>
```

```
                    <enumeration value="9967"/>
                    <enumeration value="9968"/>
                    <enumeration value="9969"/>
                    <enumeration value="9970"/>
                    <enumeration value="9971"/>
                    <enumeration value="9972"/>
                    <enumeration value="9973"/>
                    <enumeration value="9974"/>
                    <enumeration value="9975"/>
                    <enumeration value="9976"/>
                    <enumeration value="9977"/>
                    <enumeration value="9978"/>
                    <enumeration value="9979"/>
                    <enumeration value="9980"/>
                    <enumeration value="9981"/>
                    <enumeration value="9982"/>
                    <enumeration value="9983"/>
                    <enumeration value="9984"/>
                    <enumeration value="9985"/>
                    <enumeration value="9986"/>
                    <enumeration value="9987"/>
                    <enumeration value="9988"/>
                    <enumeration value="9989"/>
                    <enumeration value="9990"/>
                    <enumeration value="9991"/>
                    <enumeration value="9992"/>
                    <enumeration value="9993"/>
                    <enumeration value="9994"/>
                    <enumeration value="9995"/>
                    <enumeration value="9996"/>
                    <enumeration value="9997"/>
                    <enumeration value="9998"/>
                    <enumeration value="9999"/>
                    <!--Número da carteira inválido-->
                    <!--Número do Cartão Nacional de Saúde inválido-->
                    <!--A admissão do Beneficiário no prestador ocorreu antes da inclusão
do Beneficiário na Operadora-->
                    <!--Solicitação Anterior à Inclusão do Beneficiário-->
                    <!--Atendimento anterior à inclusão do Beneficiário-->
                    <!--Atendimento após o desligamento do Beneficiário-->
                    <!--Atendimento dentro da carência do Beneficiário-->
                    <!--Assinatura divergente-->
                    <!--Beneficiário com pagamento em aberto-->
                    <!--Assinatura do Titular / Responsável inexistente-->
                    <!--Identificação do beneficiário não consistente-->
                    <!--Serviço Profissional Hospitalar não é coberto pelo plano do
beneficiário-->
                    <!--Cadastro do beneficiário com problemas-->
                    <!--Beneficiário com data de exclusão-->
                    <!--Idade do Beneficiário acima idade limite-->
                    <!--Beneficiário com atendimento suspenso-->
                    <!--Data Validade da Carteira Vencida-->
                    <!--Empresa do beneficiário suspensa / excluída-->
                    <!--Família do Beneficiário com atendimento suspenso-->
                    <!--Via de Cartão do Beneficiário Cancelada-->
                    <!--Via de Cartão do Beneficiário não liberada-->
                    <!--Via de Cartão do beneficiário não compatível-->
                    <!--Nome do titular inválido-->
                    <!--Plano não existente-->
                    <!--Outros-->
```

<!--Quantidade de Guias informadas no Protocolo diferente das cadastradas-->
<!--Protocolo é de re-apresentação-->
<!--Protocolo não é de reapresentação-->
<!--Valor total do protocolo diferente do valor total das guias-->
<!--Outros-->
<!--Atendimento fora da vigência do contrato com o credenciado-->
<!--Número do CNES inválido-->
<!--Código Prestador inválido-->
<!--Admissão anterior à inclusão do credenciado na rede-->
<!--Admissão após o desligamento do credenciado da rede-->
<!--CPF / CNPJ inválido-->
<!--Credenciado não pertence à Rede Credenciada-->
<!--Solicitação anterior à inclusão do Credenciado-->
<!--Solicitação após o desligamento do Credenciado-->
<!--Solicitante Credenciado não cadastrado-->
<!--Assinatura / Carimbo do Credenciado inexistente-->
<!--Atendimento / Referência fora da vigência do contrato do prestador-->
<!--CBO-S (especialidade) inválido-->
<!--Credenciado não habilitado a realizar o procedimento-->
<!--Credenciado fora da abrangência geográfica do plano-->
<!--Especialidade não cadastrada-->
<!--Especialidade não cadastrada para o prestador-->
<!--Outros-->
<!--Tipo Guia inválido-->
<!--Código Tipo Guia Principal e Número Guias Incompatíveis-->
<!--Não existe o Número Guia Principal informado-->
<!--Cobrança em guia indevida-->
<!--Item pago em outra Guia-->
<!--Não existe Número Guia Principal e/ou Código Guia Principal-->
<!--Número da guia inválido-->
<!--Guia já apresentada-->
<!--Procedimento contratado não está de acordo com o tipo de guia utilizado-->
<!--Serviço do tipo cirúrgico e invasivo. Equipe médica não informada na guia-->
<!--Prestador executante não informado-->
<!--Prestador contratado não informado-->
<!--Guia com rasura-->
<!--Guia sem assinatura e/ou carimbo do credenciado. -->
<!--Guia sem data do ato cirúrgico. -->
<!--Guia com local de atendimento preenchido incorretamente. -->
<!--Guia sem data do atendimento-->
<!--Guia com código de serviço preenchido incorretamente. -->
<!--Guia sem assinatura do assistido. -->
<!--Identificação do assistido incompleta-->
<!--Validade da guia expirada-->
<!--Outros-->
<!--Acomodação não autorizada-->
<!--Procedimento não autorizado-->
<!--Não existe informação sobre a senha de autorização do procedimento-->
<!--Não existe guia de autorização relacionada-->
<!--Data de Validade da Senha é anterior a Data do Atendimento-->
<!--Número da Senha informado diferente do liberado-->
<!--Serviço solicitado não possui cobertura-->
<!--Quantidade Serviço Solicitada acima da Autorizada-->
<!--Quantidade Serviço solicitada acima coberta-->
<!--Serviço solicitado em Carência-->

<!--Solicitante não Informado-->
<!--Problemas no sistema autorizador-->
<!--Acomodação não possui cobertura-->
<!--Data de validade da senha expirada-->
<!--Procedimento não autorizado para o Beneficiário-->
<!--Solicitante não Cadastrado-->
<!--Solicitante não habilitado-->
<!--Solicitante suspenso-->
<!--Serviço solicitado já autorizado-->
<!--Serviço solicitado fora da cobertura-->
<!--Serviço solicitado é de pré-existência-->
<!--Especialidade não cadastrada para o Solicitante-->
<!--Quantidade solicitada acima da quantidade permitida-->
<!--Quantidade autorizada acima da quantidade permitida-->
<!--Necessita pré-autorização da empresa-->
<!--Não autorizado pela auditoria médica-->
<!--Necessidade de auditoria médica-->
<!--Falta de autorização da empresa de conectividade-->
<!--CBO-S (especialidade) não autorizado a realizar o serviço-->
<!--Procedimento odontológico não autorizado-->
<!--Procedimento não autorizado na face solicitada-->
<!--Procedimento não autorizado para dente/região solicitada-->
<!--Procedimento não autorizado, dente ausente-->
<!--Outros-->
<!--Tempo de Evolução da doença Inválido-->
<!--Tipo de Doença inválido-->
<!--Indicador de Acidente Inválido-->
<!--Caráter de Internação Inválido-->
<!--Regime da internação inválido-->
<!--Tipo de Internação Inválido-->
<!--Urgência não aplicável-->
<!--Código CID não Informado-->
<!--Código CID inválido-->
<!--Outros-->
<!--Reincidência no atendimento-->
<!--Tipo de atendimento inválido ou não informado-->
<!--Tipo de consulta inválido-->
<!--Tipo de saída inválido-->
<!--Intervenção anterior a Admissão-->
<!--Final da Intervenção anterior ao Início da Intervenção-->
<!--Alta Hospitalar anterior ao Final da Intervenção-->
<!--Alta anterior à data de internação-->
<!--Motivo Saída Inválido-->
<!--Óbito Mulher Inválido-->
<!--Intervenção anterior a Internação-->
<!--Serviço não pode ser realizado no local especificado-->
<!--Consulta não autorizada-->
<!--Serviço ambulatorial não autorizado-->
<!--Internação não autorizada-->
<!--Outros-->
<!--Cobrança fora do prazo de validade-->
<!--Cobrança de procedimento em duplicidade-->
<!--Horário do atendimento não está na faixa de urgência/emergência-
->
<!--Valor cobrado superior ao acordado em pacote-->
<!--Valor apresentado a maior-->
<!--Valor apresentado a menor-->
<!--Não existe informação sobre a tabela que será utilizada na
Valoração. Verifique o Contrato do Prestador-->
<!--Não existe valor para o procedimento realizado-->

<!--Falta prescrição médica-->
<!--Falta visto da Enfermagem-->
<!--Procedimento pertence a um pacote acordado e já cobrado-->
<!--Assinatura do Médico responsável pelo exame inexistente-->
<!--Faturamento Inválido-->
<!--Valor do serviço superior ao valor de tabela-->
<!--Valor do serviço inferior ao valor de tabela-->
<!--Percentual de redução/acréscimo fora dos valores definidos em tabela-->
<!--Outros-->
<!--Procedimento inválido-->
<!--Procedimento incompatível com o sexo do Beneficiário-->
<!--Idade do Beneficiário incompatível com o Procedimento-->
<!--Número de dias liberados / sessões autorizadas não informadas-->
<!--Valor total do procedimento diferente do Valor Processado-->
<!--Quantidade de procedimento deve ser maior que zero-->
<!--Procedimentos médicos duplicados-->
<!--Procedimento não conforme com CID-->
<!--Cobrança de procedimento não executado-->
<!--Cobrança de procedimento não solicitado pelo médico-->
<!--Procedimento sem registro de execução-->
<!--Cobrança de procedimento não correlacionado ao relatório específico-->
<!--Cobrança de procedimento sem justificativa para realização ou com justificativa insuficiente. -->
<!--Cobrança de procedimento com data de autorização posterior à do atendimento. -->
<!--Procedimento não autorizado-->
<!--Cobrança de procedimento em quantidade incompatível com o procedimento/evolução clínica-->
<!--Cobrança de procedimento incluso no procedimento principal-->
<!--Cobrança de procedimento que exige autorização prévia-->
<!--Cobrança de procedimento com história clínica/hipótese diagnóstica não compatível-->
<!--Cobrança de procedimento em quantidade acima da máxima permitida/autorizada-->
<!--Cobrança de procedimento não compatível com a idade. -->
<!--Cobrança de procedimento com ausência de resultado ou laudo técnico. -->
<!--Procedimento realizado pelo mesmo profissional, na mesma especialidade, no prazo inferior ao estipulado sem justificativa adequada. -->
<!--Procedimento cobrado não corresponde ao exame executado-->
<!--Cobrança de procedimento ambulatorial com data de autorização posterior à do atendimento. -->
<!--Outros-->
<!--Acomodação inválida-->
<!--Acomodação informada não está de acordo com acomodação contratada-->
<!--Permanência hospitalar incompatível com a evolução clínica-->
<!--Permanência hospitalar incompatível com o procedimento autorizado-->
<!--Quantidade de diárias deve ser maior que zero-->
<!--Acomodação não informada-->
<!--Quantidade UTI não prevista para procedimento-->
<!--Usuário não possui cobertura de UTI-->
<!--Acomodação não autorizada-->
<!--Cobrança de diárias em locais de acomodações diferentes, no mesmo dia. -->
<!--Permanência hospitalar para investigação injustificada. -->
<!--Evolução clínica não compatível com a permanência em UTI. -->

<!--Código de diária incompatível com o local de atendimento. -->
<!--Cobrança de diária em quantidade incompatível com a permanência hospitalar. -->
<!--Mudança de acomodação sem comunicação ao paciente, familiar ou acompanhante, ou sem solicitação destes. -->
<!--Cobrança de diárias de UTI incompatível com diagnóstico e evolução clínica. -->
<!--Outros-->
<!--Material inválido-->
<!--Material sem cobertura para atendimento ambulatorial-->
<!--Material não especificado-->
<!--Material sem nota fiscal do fornecedor-->
<!--Quantidade de material deve ser maior que zero-->
<!--Material informado não coberto-->
<!--Cobrança de material em quantidade incompatível com a permanência. -->
<!--Cobrança de material em quantidades incompatíveis com o procedimento realizado. -->
<!--Quantidade de material superior a quantidade coberta-->
<!--Cobrança de materiais inclusos nas taxas-->
<!--Cobrança de material incluso no pacote negociado. -->
<!--Cobrança de material incompatível com o relatório técnico. -->
<!--Cobrança de material em permanência hospitalar não autorizada. -->
<!--Cobrança de material não utilizado-->
<!--Outros-->
<!--Medicamento inválido-->
<!--Medicamento sem cobertura para atendimento ambulatorial-->
<!--Medicamento não especificado-->
<!--Medicamento sem nota fiscal do fornecedor-->
<!--Quantidade de medicamentos deve ser maior que zero-->
<!--Medicamento informado não coberto-->
<!--Cobrança de medicamento em quantidade incompatível com a permanência. -->
<!--Cobrança de medicamento em quantidades incompatíveis com o procedimento realizado. -->
<!--Quantidade de medicamento superior a quantidade coberta-->
<!--Cobrança de medicamento inclusos nas taxas-->
<!--Cobrança de medicamento incluso no pacote negociado. -->
<!--Cobrança de medicamento incompatível com o relatório técnico. -->
<!--Cobrança de medicamento em permanência hospitalar não autorizada. -->
<!--Cobrança de medicamento não utilizado-->
<!--Outros-->
<!--OPM inválido-->
<!--OPM sem cobertura para atendimento ambulatorial-->
<!--OPM sem nota fiscal do fornecedor-->
<!--Quantidade de OPM deve ser maior que zero-->
<!--OPM informado não coberto-->
<!--OPM informado não autorizado-->
<!--Cobrança de OPM não utilizado-->
<!--Cobrança de OPM no item material e medicamentos. -->
<!--Cobrança de OPM em desacordo com relatório técnico-->
<!--Cobrança de OPM em quantidade incompatível com o procedimento realizado-->
<!--Cobrança de OPM inclusa no pacote-->
<!--Outros-->
<!--Gases Medicinais inválidos-->
<!--Cobrança de oxigenoterapia sem prescrição médica. -->

<!--Cobrança de oxigenoterapia com quantitativo de uso em divergência/pago valor corrigido. -->
<!--Cobrança de oxigênio incluso na taxa de nebulização especificada. -->
<!--Cobrança de oxigenoterapia em uso prolongado sem justificativa de uso. -->
<!--Cobrança de oxigenoterapia sem registro de controle de uso (entrada e saída). -->
<!--Cobrança de gases em quantidade superior ao período de permanência-->
<!--Cobrança de CO2 nas cirurgias videolaparoscópicas durante toda a realização do procedimento (início ao fim). -->
<!--Cobrança de ar comprimido sem registro no Boletim Anestésico e duração de uso. -->
<!--Cobrança de gases incompatível com o utilizado/ prescrito. -->
<!--Outros-->
<!--Taxa / Aluguel inválido-->
<!--Cobrança de taxa por uso de equipamento incompatível com o procedimento realizado/uso previsto no procedimento. -->
<!--Cobrança de taxa de uso de bomba de infusão em paciente internado na UTI-->
<!--Cobrança de outras taxas associadas/inclusas na cobrança da taxa de sala prevista. -->
<!--Cobrança de mais de uma taxa de sala de cirurgia, por conta do número de procedimentos realizados no mesmo tempo cirúrgico. -->
<!--Cobrança indevida de taxa de sala por administração de medicamentos. -->
<!--Cobrança de taxas, de serviços realizados em ambientes incompatíveis com o uso de equipamentos. -->
<!--Cobrança de taxas em quantidade superior ao tempo de permanência hospitalar-->
<!--Cobrança de taxa de observação em pronto socorro com permanência menor que o período estipulado-->
<!--Cobrança de taxa de observação em pronto socorro sem o registro da permanência. -->
<!--Cobrança de taxa de sala de pronto socorro, para aplicação de medicamentos. -->
<!--Cobrança de taxa de recuperação anestésica não justificada para o procedimento. -->
<!--Cobrança de taxa inclusa no pacote negociado. -->
<!--Cobrança de taxa de equipamento em concomitância com a cobrança de taxa para o procedimento. -->
<!--Taxa exige informação do valor na Guia. -->
<!--Cobrança de taxa de recuperação anestésica para pacientes com pós-operatório imediato realizado na UTI/CTI. -->
<!--Cobrança de taxa de recuperação anestésica sem a presença do anestesista. -->
<!--Cobrança de taxa de sala incompatível com o procedimento. -->
<!--Cobrança de taxa de observação para atendimento que gerou uma internação. -->
<!--Cobrança de taxa de sala cirúrgica com porte anestésico diferente do procedimento autorizado/realizado. -->
<!--Cobrança de taxa em quantidade incorreta. -->
<!--Cobrança de taxa por uso de equipamento de uso obrigatório na sala de cirurgia, cuja taxa de sala cirúrgica já inclui seu uso. -->
<!--Cobrança de taxa de equipamentos de uso obrigatório no local de atendimento. -->
<!--Cobrança de taxa de observação para atendimento que gerou uma internação. -->
<!--Outros-->

<!--Procedimento em série inválido-->
<!--Cobrança de duas avaliações fisioterápicas-->
<!--Cobrança de psicoterapia individual, quando o aplicado é a cobrança de psicoterapia em grupo-->
<!--Quantidade de sessões cobradas não condizem com as assinaturas no controle de tratamento seriado-->
<!--O código cobrado é diferente do código autorizado-->
<!--A quantidade de sessões cobradas é diferente da quantidade autorizada-->
<!--O código autorizado está incompatível com a prescrição médica solicitada-->
<!--Cobrança de sessões sem o devido plano de tratamento e, ou, com o prazo de pagamento expirado-->
<!--Cobrança do procedimento seriado incompatível com o quadro clínico-->
<!--Cobrança do procedimento seriado em número de sessões acima da quantidade estabelecida-->
<!--Ausência de evolução no prontuário médico do tratamento seriado realizado. -->
<!--Cobrança de sessões de fisioterapia em desacordo com as evoluções do prontuário médico-->
<!--Cobrança de tratamento seriado sem justificativa clínica/técnica-->
<!--Serviço não contratado para o prestador-->
<!--Local de atendimento inadequado-->
<!--Quantidade cobrada diferente da realizada-->
<!--Outros-->
<!--Codificação incorreta/inadequada do procedimento. -->
<!--Cobrança de honorário incluso no procedimento principal-->
<!--Cobrança de honorário sem registro da efetiva participação do profissional-->
<!--Procedimento principal não requer equipe médica-->
<!--Não cabe pagamento do honorário integral por ser a mesma via de acesso cirúrgico. -->
<!--Cobrança do honorário em local de atendimento incorreto (inexistente). -->
<!--Cobrança de Honorários em duplicidade.-->
<!--Cobrança de consulta indevida, quando o procedimento principal já está sendo remunerado. -->
<!--Local de atendimento não informado. -->
<!--Outros-->
<!--Procedimento inválido-->
<!--Cobrança de exame não solicitado pelo médico-->
<!--Exame sem registro de execução-->
<!--Cobrança de exame não correlacionado ao relatório específico-->
<!--Cobrança de procedimento/exame sem justificativa para realização ou com justificativa insuficiente. -->
<!--Cobrança de procedimento/exame com data de autorização posterior à do atendimento. -->
<!--Exame não autorizado-->
<!--Cobrança de exame em quantidade incompatível com o procedimento/evolução clínica-->
<!--Cobrança de procedimento incluso no procedimento principal-->
<!--Cobrança de exame que exige autorização prévia-->
<!--Cobrança de exame com história clínica/hipótese diagnóstica não compatível-->
<!--Cobrança de exame em quantidade acima da máxima permitida/autorizada-->
<!--Cobrança de exame não compatível com a idade. -->
<!--Cobrança de exame com ausência de resultado ou laudo técnico. -->

<!--Exame realizado pelo mesmo profissional, na mesma especialidade, no prazo inferior ao estipulado sem justificativa adequada. -->
<!--Exame cobrado não corresponde ao exame executado-->
<!--Cobrança de exame ambulatorial com data de autorização posterior à do atendimento. -->
<!--Exames não justificam caráter de urgência-->
<!--Outros-->
<!--Pacote inválido-->
<!--Pacote incompatível com o sexo do Beneficiário-->
<!--Idade do Beneficiário incompatível com o Pacote-->
<!--Valor total do pacote diferente do Valor Processado-->
<!--Valor do pacote superior ao valor dos itens-->
<!--Cobrança de pacote não executado-->
<!--Cobrança de pacote não solicitado pelo médico-->
<!--Pacote sem registro de execução-->
<!--Cobrança de pacote não correlacionado ao relatório especifico-->
<!--Cobrança de pacote sem justificativa para realização ou com justificativa insuficiente. -->
<!--Cobrança de pacote com data de autorização posterior à do atendimento. -->
<!--Pacote não autorizado-->
<!--Cobrança de pacote em quantidade incompatível com o procedimento/evolução clínica-->
<!--Itens de composição do pacote não realizados-->
<!--Cobrança do pacote exige autorização prévia-->
<!--Cobrança de pacote com história clínica/hipótese diagnóstica não compatível-->
<!--Cobrança de pacote em quantidade acima da máxima permitida/autorizada-->
<!--Cobrança de pacote não compatível com a idade. -->
<!--Cobrança de pacote com ausência de resultado ou laudo técnico. -->
<!--Pacote realizado pelo mesmo profissional, na mesma especialidade, no prazo inferior ao estipulado sem justificativa adequada. -->
<!--Pacote cobrado não corresponde ao exame executado-->
<!--Cobrança de pacote ambulatorial com data de autorização posterior à do atendimento. -->
<!--Outros-->
<!--Revisão de glosa inválida-->
<!--Glosa mantida-->
<!--Pedido de revisão sem justificativa-->
<!--Mais de um recurso de glosa para a mesma guia/protocolo-->
<!--A guia não é de revisão-->
<!--Número da guia inválido-->
<!--Outros-->
<!--Procedimento odontológico inválido-->
<!--Cobrança de procedimento odontológico que exige autorização prévia-->
<!--Idade do Beneficiário incompatível com o Procedimento odontológico-->
<!--Cobrança de procedimento odontológico em quantidade acima da máxima permitida/autorizada-->
<!--Valor total do procedimento diferente do Valor Processado-->
<!--Quantidade de procedimento deve ser maior que zero-->
<!--Procedimentos odontológicos duplicados-->
<!--Cobrança de procedimento odontológico incluso no procedimento principal-->
<!--Cobrança de procedimento odontológico não executado-->
<!--Cobrança de procedimento não solicitado pelo cirurgião-dentista-->
<!--Procedimento odontológico sem registro de execução-->

<!--Cobrança de procedimento odontológico não correlacionado ao relatório específico-->
<!--Cobrança de procedimento odontológico sem justificativa para realização ou com justificativa insuficiente. -->
<!--Cobrança de procedimento odontológico com data de autorização posterior à do atendimento. -->
<!--Cobrança de procedimento odontológico com ausência de resultado ou laudo técnico. -->
<!--Procedimento odontológico realizado pelo mesmo profissional, na mesma especialidade, no prazo inferior ao estipulado sem justificativa adequada. -->
<!--Procedimento cobrado não corresponde a perícia (especificar). -->
<!--Evento glosado por auditoria (especificar) -->
<!--Evento sob análise técnica, aguardando liberação de confirmação para posterior pagamento-->
<!--Conforme documentação radiográfica enviada, evento realizado inadequadamente-->
<!--Falha em informação de dados de arcadas/hemi-arcos-->
<!--Falha em informação de dados de dente inicial e/ou final-->
<!--Falha em informação de dados de faces dos dentes-->
<!--Evento só possível em dentes decíduos-->
<!--Evento só possível em dentes permanentes-->
<!--Erro nas informações de ordem dos dentes inicial e final-->
<!--Desacordo entre o tipo de dente e o número de canais solicitados-->
<!--Evento restrito à especialistas-->
<!--Evento não indicado pela Auditoria Inicial-->
<!--Auditoria final consta que a restauração foi realizada em outro material-->
<!--RX fora dos padrões técnicos-->
<!--Intervalo da última MPP inferior a três meses-->
<!--Intervalo da última MPP inferior a quatro meses-->
<!--Justificativa tecnicamente não satisfatória-->
<!--Paciente em tratamento com o mesmo profissional-->
<!--Paciente em tratamento com outro profissional-->
<!--Procedimento cobrado não é igual ao executado-->
<!--RX inicial incongruente com o RX final-->
<!--RX não corresponde ao procedimento cobrado-->
<!--Glosa técnica (especificar detalhadamente) -->
<!--Outros-->
<!--Mensagem eletrônica fora do padrão TISS-->
<!--Não foi possível validar o arquivo XML-->
<!--Endereço do Remetente inválido-->
<!--Endereço do destinatário inválido-->
<!--Remetente não identificado-->
<!--Destinatário não identificado-->
<!--Mensagem inconsistente ou incompleta-->
<!--Espaço reservado para a caixa de saída insuficiente-->
<!--Espaço reservado para a caixa de entrada insuficiente-->
<!--Envio de mensagem não foi terminado-->
<!--Envio de mensagem finalizada-->
<!--Recebimento de mensagem não finalizado-->
<!--Recebimento de mensagem finalizada-->
<!--Código Hash inválido. Mensagem pode estar corrompida. -->
<!--Número de guias/demonstrativos dentro da mensagem superior ao tamanho máximo permitido.-->
<!--Erro não identificado-->
<!--Mensagens customizaveis-->
<!--Mensagens customizaveis-->
<!--Mensagens customizaveis-->
<!--Mensagens customizaveis-->

<!--Mensagens customizaveis-->

<!--Mensagens customizaveis-->
<!--Mensagens customizaveis-->
<!--Mensagens customizaveis-->

```
                <!--Mensagens customizaveis-->
                <!--Mensagens customizaveis-->
                <!--Mensagens customizaveis-->
```

```
<enumeration value="010"/>
<enumeration value="011"/>
<enumeration value="012"/>
<enumeration value="013"/>
<enumeration value="014"/>
<enumeration value="015"/>
<enumeration value="016"/>
<enumeration value="017"/>
<enumeration value="019"/>
<enumeration value="020"/>
<enumeration value="021"/>
<enumeration value="022"/>
<enumeration value="023"/>
<enumeration value="024"/>
<enumeration value="027"/>
<enumeration value="028"/>
<enumeration value="030"/>
<enumeration value="031"/>
<enumeration value="032"/>
<enumeration value="033"/>
<enumeration value="036"/>
<enumeration value="037"/>
<enumeration value="038"/>
<enumeration value="039"/>
<enumeration value="040"/>
<enumeration value="043"/>
<enumeration value="045"/>
<enumeration value="046"/>
<enumeration value="050"/>
<enumeration value="052"/>
<enumeration value="053"/>
<enumeration value="054"/>
<enumeration value="055"/>
<enumeration value="056"/>
<enumeration value="059"/>
<enumeration value="060"/>
<enumeration value="062"/>
<enumeration value="063"/>
<enumeration value="064"/>
<enumeration value="065"/>
<enumeration value="067"/>
<enumeration value="070"/>
<enumeration value="071"/>
<enumeration value="072"/>
<enumeration value="073"/>
<enumeration value="074"/>
<enumeration value="076"/>
<enumeration value="077"/>
<enumeration value="079"/>
<enumeration value="081"/>
<enumeration value="082"/>
<enumeration value="087"/>
<enumeration value="088"/>
<enumeration value="089"/>
<enumeration value="090"/>
<enumeration value="091"/>
<enumeration value="092"/>
<enumeration value="094"/>
<enumeration value="095"/>
<enumeration value="096"/>
```

```
<enumeration value="097"/>
<enumeration value="098"/>
<enumeration value="099"/>
<enumeration value="100"/>
<enumeration value="101"/>
<enumeration value="103"/>
<enumeration value="104"/>
<enumeration value="105"/>
<enumeration value="106"/>
<enumeration value="108"/>
<enumeration value="452"/>
<enumeration value="453"/>
<enumeration value="465"/>
<enumeration value="468"/>
<enumeration value="469"/>
<enumeration value="470"/>
<enumeration value="471"/>
<enumeration value="472"/>
<enumeration value="473"/>
<enumeration value="474"/>
<enumeration value="475"/>
<enumeration value="476"/>
<enumeration value="477"/>
<enumeration value="478"/>
<enumeration value="479"/>
<enumeration value="480"/>
<enumeration value="481"/>
<enumeration value="482"/>
<enumeration value="483"/>
<enumeration value="484"/>
<enumeration value="485"/>
<enumeration value="486"/>
<enumeration value="487"/>
<enumeration value="495"/>
<enumeration value="496"/>
<enumeration value="497"/>
<enumeration value="498"/>
<enumeration value="499"/>
<enumeration value="500"/>
<enumeration value="501"/>
<enumeration value="502"/>
<enumeration value="503"/>
<enumeration value="504"/>
<enumeration value="505"/>
<enumeration value="506"/>
<enumeration value="507"/>
<enumeration value="508"/>
<enumeration value="509"/>
<enumeration value="510"/>
<enumeration value="511"/>
<enumeration value="512"/>
<enumeration value="513"/>
<enumeration value="514"/>
<enumeration value="515"/>
<enumeration value="516"/>
<enumeration value="517"/>
<enumeration value="518"/>
<enumeration value="519"/>
<enumeration value="520"/>
<enumeration value="521"/>
```

```
<enumeration value="522"/>
<enumeration value="523"/>
<enumeration value="524"/>
<enumeration value="525"/>
<enumeration value="526"/>
<enumeration value="527"/>
<enumeration value="528"/>
<enumeration value="529"/>
<enumeration value="530"/>
<enumeration value="531"/>
<enumeration value="532"/>
<enumeration value="533"/>
<enumeration value="534"/>
<enumeration value="535"/>
<enumeration value="536"/>
<enumeration value="537"/>
<enumeration value="538"/>
<enumeration value="539"/>
<enumeration value="540"/>
<enumeration value="541"/>
<enumeration value="542"/>
<enumeration value="543"/>
<enumeration value="544"/>
<enumeration value="545"/>
<enumeration value="546"/>
<enumeration value="547"/>
<enumeration value="548"/>
<enumeration value="549"/>
<enumeration value="550"/>
<enumeration value="551"/>
<enumeration value="552"/>
<enumeration value="553"/>
<enumeration value="554"/>
<enumeration value="555"/>
<enumeration value="556"/>
<enumeration value="557"/>
<enumeration value="558"/>
<enumeration value="559"/>
<enumeration value="560"/>
<enumeration value="561"/>
<enumeration value="562"/>
<enumeration value="563"/>
<enumeration value="564"/>
<enumeration value="565"/>
<enumeration value="566"/>
<enumeration value="567"/>
<enumeration value="568"/>
<enumeration value="569"/>
<enumeration value="570"/>
<enumeration value="571"/>
<enumeration value="572"/>
<enumeration value="573"/>
<enumeration value="645"/>
<enumeration value="646"/>
<enumeration value="650"/>
<enumeration value="651"/>
<enumeration value="652"/>
<enumeration value="653"/>
<enumeration value="654"/>
<!--Acesso-->
```

<!--Adro-->
<!--Alameda-->
<!--Alto-->
<!--Atalho-->
<!--Avenida-->
<!--Balneário-->
<!--Belvedere-->
<!--Beco-->
<!--Bloco-->
<!--Bosque-->
<!--Boulevard-->
<!--Baixa-->
<!--Cais-->
<!--Caminho-->
<!--Chapadão-->
<!--Conjunto-->
<!--Colônia-->
<!--Corredor-->
<!--Campo-->
<!--Córrego-->
<!--Desvio-->
<!--Distrito-->
<!--Escada-->
<!--Estrada-->
<!--Estação-->
<!--Estádio-->
<!--Favela-->
<!--Fazenda-->
<!--Ferrovia-->
<!--Fonte-->
<!--Feira-->
<!--Forte-->
<!--Galeria-->
<!--Granja-->
<!--Ilha-->
<!--Jardim-->
<!--Ladeira-->
<!--Largo-->
<!--Lagoa-->
<!--Loteamento-->
<!--Morro-->
<!--Monte-->
<!--Paralela-->
<!--Passeio-->
<!--Pátio-->
<!--Praça-->
<!--Parada-->
<!--Praia-->
<!--Prolongamento-->
<!--Parque-->
<!--Passarela-->
<!--Passagem-->
<!--Ponte-->
<!--Quadra-->
<!--Quinta-->
<!--Rua-->
<!--Ramal-->
<!--Recanto-->
<!--Retiro-->
<!--Reta-->

<!--Rodovia-->
<!--Retorno-->
<!--Sítio-->
<!--Servidão-->
<!--Setor-->
<!--Subida-->
<!--Trincheira-->
<!--Terminal-->
<!--Trevo-->
<!--Travessa-->
<!--Via-->
<!--Viaduto-->
<!--Vila-->
<!--Viela-->
<!--Vale-->
<!--Zigue-zague-->
<!--Trecho-->
<!--Vereda-->
<!--Artéria-->
<!--Elevada-->
<!--Porto-->
<!--Balão-->
<!--Paradouro-->
<!--Área-->
<!--Jardinete-->
<!--Esplanada-->
<!--Quintas-->
<!--Rotula-->
<!--Marina-->
<!--Descida-->
<!--Circular-->
<!--Unidade-->
<!--Chácara-->
<!--Rampa-->
<!--Ponta-->
<!--Via de pedestre-->
<!--Condomínio-->
<!--Habitacional-->
<!--Residencial-->
<!--Canal-->
<!--Buraco-->
<!--Módulo-->
<!--Estância-->
<!--Lago-->
<!--Núcleo-->
<!--Aeroporto-->
<!--Passagem Subterrânea-->
<!--Complexo Viário-->
<!--Praça de Esportes-->
<!--Via Elevada-->
<!--Rotatória-->
<!--1ª Travessa-->
<!--2ª Travessa-->
<!--3ª Travessa-->
<!--4ª Travessa-->
<!--5ª Travessa-->
<!--6ª Travessa-->
<!--7ª Travessa-->
<!--8ª Travessa-->
<!--9ª Travessa-->

<!--10ª Travessa-->
<!--11ª Travessa-->
<!--12ª Travessa-->
<!--13ª Travessa-->
<!--14ª Travessa-->
<!--15ª Travessa-->
<!--16ª Travessa-->
<!--1º Alto-->
<!--2º Alto-->
<!--3º Alto-->
<!--4º Alto-->
<!--5º Alto-->
<!--1º Beco-->
<!--2º Beco-->
<!--3º Beco-->
<!--4º Beco-->
<!--5º Beco-->
<!--1ª Paralela-->
<!--2ª Paralela-->
<!--3ª Paralela-->
<!--4ª Paralela-->
<!--5ª Paralela-->
<!--1ª Subida-->
<!--2ª Subida-->
<!--3ª Subida-->
<!--4ª Subida-->
<!--5ª Subida-->
<!--6ª Subida-->
<!--1ª Vila-->
<!--2ª Vila-->
<!--3ª Vila-->
<!--4ª Vila-->
<!--5ª Vila-->
<!--1º Parque-->
<!--2º Parque-->
<!--3º Parque-->
<!--1ª Rua-->
<!--2ª Rua-->
<!--3ª Rua-->
<!--4ª Rua-->
<!--5ª Rua-->
<!--6ª Rua-->
<!--7ª Rua-->
<!--8ª Rua-->
<!--9ª Rua-->
<!--10ª Rua-->
<!--11ª Rua-->
<!--12ª Rua-->
<!--Estacionamento-->
<!--Vala-->
<!--Rua de Pedestre-->
<!--Túnel-->
<!--Variante-->
<!--Rodo Anel-->
<!--Travessa Particular-->
<!--Calçada -->
<!--Via de Acesso-->
<!--Entrada Particular-->
<!--Acampamento-->
<!--Via Expressa-->

```
                <!--Estrada Municipal-->
                <!--Avenida Contorno-->
                <!--Entre-quadra-->
                <!--Rua de Ligação-->
                <!--Área Especial-->
            </restriction>
        </simpleType>
        <simpleType name="st_tipoSaidaGuiaConsulta">
            <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <enumeration value="4"/>
                <enumeration value="5"/>
                <!--Retorno-->
                <!--Retorno SADT-->
                <!--Referência-->
                <!--Internação-->
                <!--Alta-->
            </restriction>
        </simpleType>
        <simpleType name="st_tipoSaidaGuiaSADT">
            <restriction base="string">
                <enumeration value="1"/>
                <enumeration value="2"/>
                <enumeration value="3"/>
                <enumeration value="4"/>
                <enumeration value="5"/>
                <enumeration value="6"/>
                <!--Retorno-->
                <!--Retorno SADT-->
                <!--Referência-->
                <!--Internação-->
                <!--Alta-->
                <!--Óbito-->
            </restriction>
        </simpleType>
        <simpleType name="st_tipoTransacao">
            <restriction base="string">
                <enumeration value="ENVIO_LOTE_GUIAS"/>
                <enumeration value="SOLIC_DEMONSTRATIVO_RETORNO"/>
                <enumeration value="SOLIC_STATUS_PROTOCOLO"/>
                <enumeration value="SOLICITACAO_PROCEDIMENTOS"/>
                <enumeration value="PROTOCOLO_RECEBIMENTO"/>
                <enumeration value="SITUACAO_PROTOCOLO"/>
                <enumeration value="RESPOSTA_SOLICITACAO"/>
                <enumeration value="DEMONSTRATIVO_PAGAMENTO"/>
                <enumeration value="DEMONSTRATIVO_ANALISE_CONTA"/>
                <enumeration value="DEMONSTRATIVO_ODONTOLOGIA"/>
                <enumeration value="CANCELAMENTO_GUIA_RECIBO"/>
                <enumeration value="RE_APRESENTACAO_GUIA"/>
                <enumeration value="CANCELA_GUIA"/>
                <enumeration value="VERIFICA_ELEGIBILIDADE"/>
                <enumeration value="SITUACAO_ELEGIBILIDADE"/>
                <enumeration value="AUTORIZACAO_ODONTOLOGIA"/>
                <enumeration value="SOLICITA_STATUS_AUTORIZACAO"/>
                <enumeration value="STATUS_AUTORIZACAO"/>
            </restriction>
        </simpleType>
        <simpleType name="st_UF">
```

```
            <restriction base="string">
                <length value="2"/>
                <enumeration value="RS"/>
                <enumeration value="SC"/>
                <enumeration value="PR"/>
                <enumeration value="SP"/>
                <enumeration value="MG"/>
                <enumeration value="RJ"/>
                <enumeration value="ES"/>
                <enumeration value="MS"/>
                <enumeration value="MT"/>
                <enumeration value="GO"/>
                <enumeration value="TO"/>
                <enumeration value="PA"/>
                <enumeration value="AM"/>
                <enumeration value="RO"/>
                <enumeration value="RR"/>
                <enumeration value="AC"/>
                <enumeration value="DF"/>
                <enumeration value="BA"/>
                <enumeration value="SE"/>
                <enumeration value="CE"/>
                <enumeration value="PI"/>
                <enumeration value="PB"/>
                <enumeration value="RN"/>
                <enumeration value="AL"/>
                <enumeration value="MA"/>
                <enumeration value="PE"/>
                <enumeration value="AP"/>
                <!--Rio Grande do Sul-->
                <!--Santa Catarina-->
                <!--Paraná-->
                <!--São Paulo-->
                <!--Minas Gerais-->
                <!--Rio de Janeiro-->
                <!--Espírito Santo-->
                <!--Mato Grosso do Sul-->
                <!--Mato Grosso-->
                <!--Goiás-->
                <!--Tocantins-->
                <!--Pará-->
                <!--Amazonas-->
                <!--Rondônia-->
                <!--Roraima-->
                <!--Acre-->
                <!--Distrito Federal-->
                <!--Bahia-->
                <!--Sergipe-->
                <!--Ceará-->
                <!--Piauí-->
                <!--Paraíba-->
                <!--Rio Grande do Norte-->
                <!--Alagoas-->
                <!--Maranhão-->
                <!--Pernambuco-->
                <!--Amapá-->
            </restriction>
        </simpleType>
        <simpleType name="st_unidadeTempo">
            <restriction base="string">
```

```
                        <length value="1"/>
                        <enumeration value="A"/>
                        <enumeration value="M"/>
                        <enumeration value="D"/>
                        <!--Anos-->
                        <!--Meses-->
                        <!--Dias-->
                </restriction>
        </simpleType>
        <simpleType name="st_valorMonetario">
                <restriction base="decimal">
                        <fractionDigits value="2"/>
                        <totalDigits value="10"/>
                </restriction>
        </simpleType>
        <simpleType name="st_versao">
                <restriction base="string">
                        <enumeration value="2.01.01"/>
                        <enumeration value="2.01.02"/>
                        <enumeration value="2.01.03"/>
                </restriction>
        </simpleType>
        <simpleType name="st_viaDeAcesso">
                <restriction base="string">
                        <maxLength value="1"/>
                        <enumeration value="U"/>
                        <enumeration value="M"/>
                        <enumeration value="D"/>
                        <!--Única-->
                        <!--Mesma via-->
                        <!--Diferentes vias-->
                </restriction>
        </simpleType>
</schema>
```

26. CONCLUSÃO

A implantação de sistemas informatizados para a área de saúde auxiliará na comunicação e rapidez nos processos que estão sendo implementados pelo padrão.

A adaptação de prestadores e operadoras ao novo modelo é exigida de maneira obrigatória assim, todos os envolvidos devem se adequar a nova normatização.

A intenção desse projeto foi propiciar um sistema com interface amigável e de fácil entendimento, possibilitando assim, uma melhor aceitação a nova mudança.

REFERENCIAS

AGÊNCIA NACIONAL DE SAÚDE<http://ans.gov.br> Acesso desde de maio de 2008 até dia 07/12/2008

APACHE ORG. Disponível em <HTTP://xeerce.apache.org/xerces-j/> Acesso em 01/11/2008.

BASHAM, Crian; Sierra, Kathy; Bates, Bert – Use a Cabeça! Servlets & JSP – Rio de Janeiro: Editora Alta Books Ltda., 2005

BECKER, Fernando; FARINA, Sérgio; SCHEID, Urbano. Apresentação de trabalhos escolares: Orientação para datilografia e digitação. Porto Alegre: Multilivro, 2000.

CRIARWEB. Disponível em <HTTP://www.criarweb.com.br> Acesso em: 25/09/2008.

DEITEL, H.M. – Java: como programar / 6.ed. – São Paulo: Pearson Education do Brasil, 2005

DESTROYDROP. Disponível em <http://destroydrop.com/javascripts/tree/> Acesso em 07/09/2008

EMBLEMATIQ. Disponível em
<http://www.emblematiq.com/projects/niceforms/>
Acesso em 05/12/2008

FREEMAN, Elisabeth; Freeman, Eric – Use a Cabeça! HTML com CSS & XHTML – Rio de Janeiro: Editora Alta Books Ltda., 2006

GUERRA, Isabel Cristina Vilela. Profissão: "Administrador". Disponível em: <http://www.unicaldas.edu.br/paginas/administracao/prof_admistrador.pdf>. Acesso em: 14 maio 2008.

GUEDES, Gilleanes T.A. – UML2 Guia Prático. – São Paulo: Novatec Editora, 2007.

J2EEBRASIL. Disponível em <HTTP://www.j2eebrasil.com.br> Acesso em 21/10/2008

JUDE. Disponível em <http://jude.change-vision.com/jude-web/product/community.html> Acesso em 20/11/2008

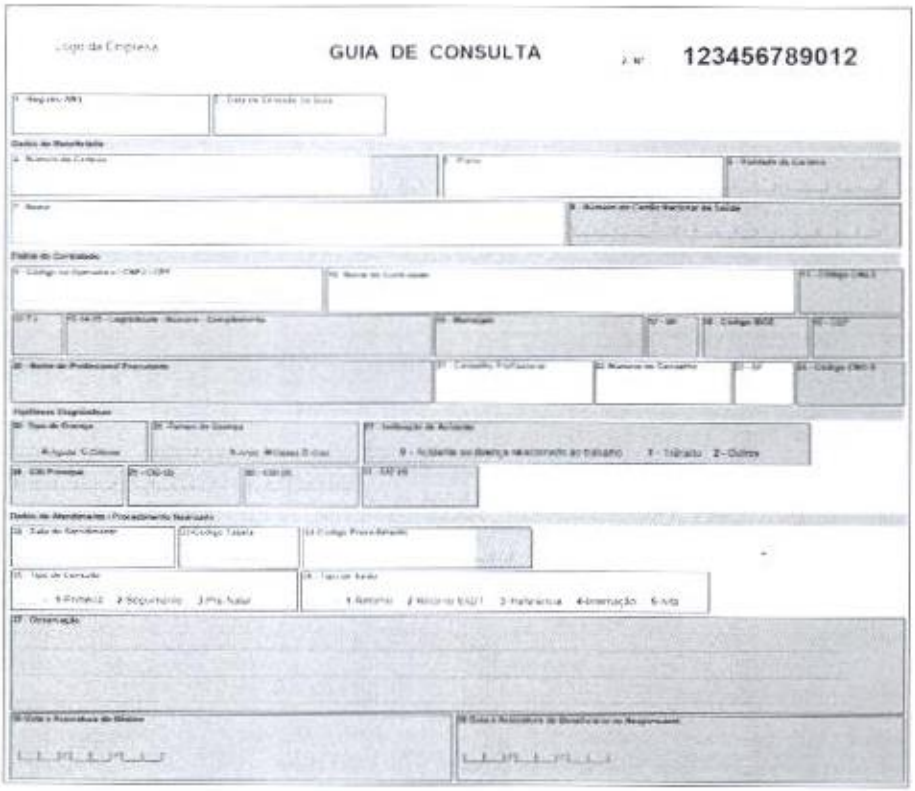
KURNIAWAN, Budi – Java para a Web com Servlets, JSP e EJB. – Rio de Janeiro: Editora Ciência Moderna Ltda., 2002.

SILVA, Mary Aparecida Ferreira – Métodos e Técincas de Pesquisa. 2.ed. ver.atual/ Mary Aparecida da Silva – Curitiba: Ibpex, 2005

THUMS, Jorge. Acesso à realidade: Técnicas de pesquisa e construção do conhecimento. Porto Alegre: Sulina/Ulbra, 2000.

WORL WIDE WEB CONSORTIUM <http://www.w3.org>Acesso em 04/10/2008

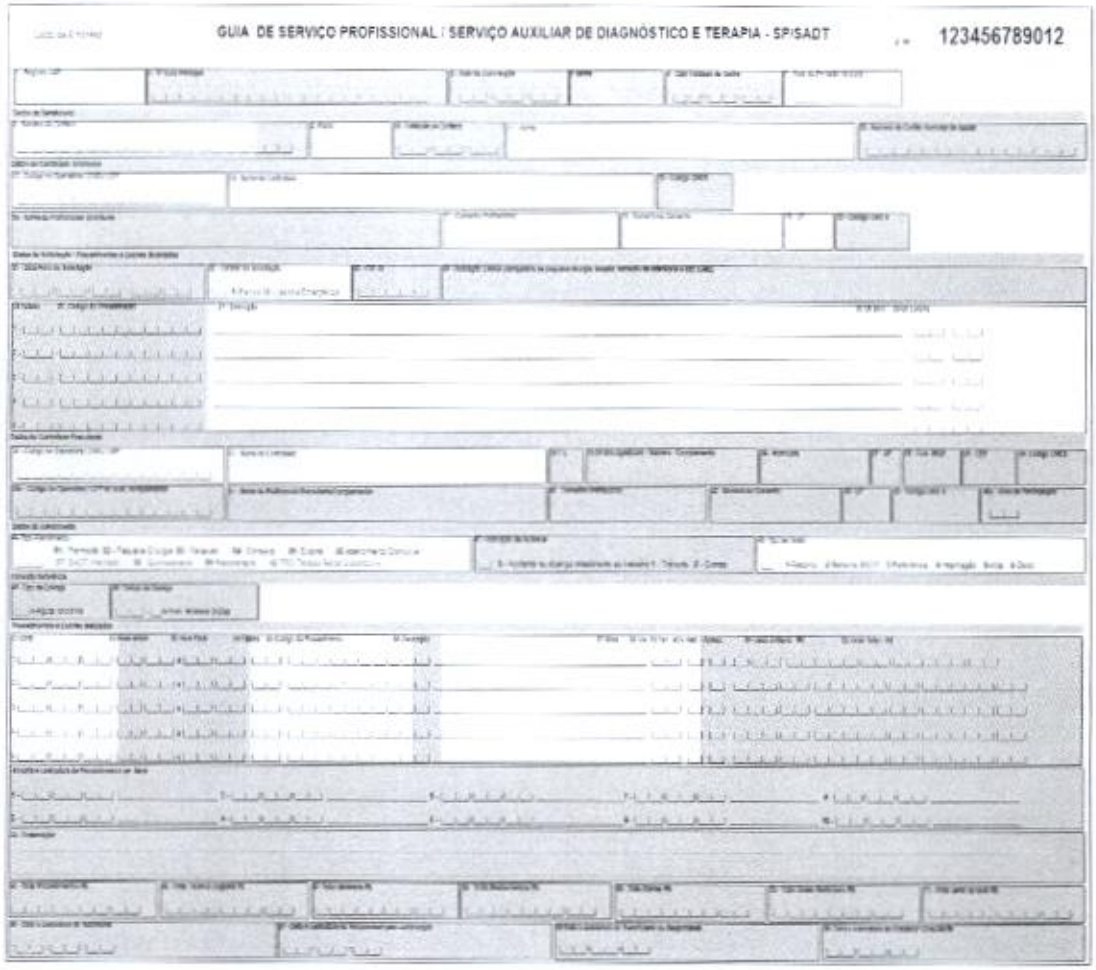
ANEXOS



Logo da Empresa

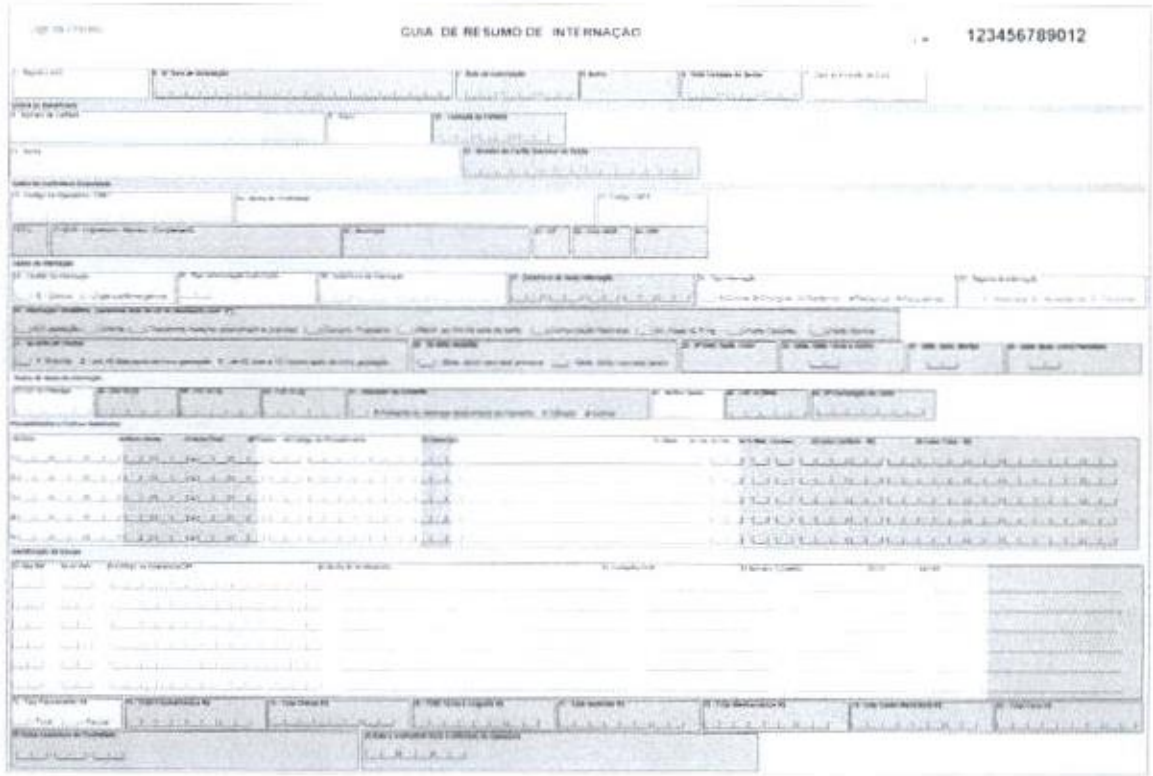
GUIA DE CONSULTA

Nº 123456789012



GUIA DE SERVIÇO PROFISSIONAL / SERVIÇO AUXILIAR DE DIAGNÓSTICO E TERAPIA - SP/SADT

123456789012



GUIA DE RESUMO DE INTERNAÇÃO

123456789012

Logo da Empresa

GUIA DE SOLICITAÇÃO DE INTERNAÇÃO

2 Nº 123456789012

1 - Registro ANS | 2 - Data da Autorização | 4 - Senha | 5 - Data Validade da Senha | 6 - Data de Emissão da Guia

Dados do Beneficiário

7 - Número da Carteira | 8 - Plano | 9 - Validade da Carteira

10 - Nome | 11 - Número do Cartão Nacional de Saúde

Dados do Contratado Solicitante

12 - Código na Operadora / CNPJ / CPF | 13 - Nome do Contratado | 14 - Código CNES

15 - Nome do Profissional Solicitante | 16 - Conselho Profissional | 17 - Número no Conselho | 18 - UF | 19 - Código CBO S

Dados do Contratado Solicitado / Dados da Internação

20 - Código na Operadora / CNPJ | 21 - Nome do Prestador

22 - Caráter da Internação (1 Eletiva 2 Urgência/Emergência) | 23 - Tipo de Internação (1 - Clínica 2 - Cirúrgica 3 - Obstetrícia 4 - Pediátrica 5 - Psiquiátrica)

24 - Regime de Internação (1 - Hospitalar 2 - Hospital-dia 3 - Domiciliar) | 25 - Qtde. Diárias Solicitadas

26 - Indicação Clínica

Hipóteses Diagnósticas

27 - Tipo Doença (A - Aguda C - Crônica) | 28 - Tempo da Doença Referido pelo Paciente (A-Anos M-Meses D-Dias) | 29 - Indicação de Acidente (0 - Acidente ou doença relacionada ao Trabalho 1 - Trânsito 2 - Outros)

30 - CID 10 Principal | 31 - CID 10 (2) | 32 - CID 10 (3) | 33 - CID 10 (4)

Procedimentos Solicitados

34 - Tabela	35 - Código do Procedimento	36 - Descrição	37 - Qtde. Solic.	38 - Qtde. Aut.
1				
2				
3				
4				
5				

OPM Solicitados

39 - Tabela	40 - Código do OPM	41 - Descrição OPM	42 - Qtde.	43 - Fabricante	44 - Valor Unitário R$
1					
2					
3					
4					
5					

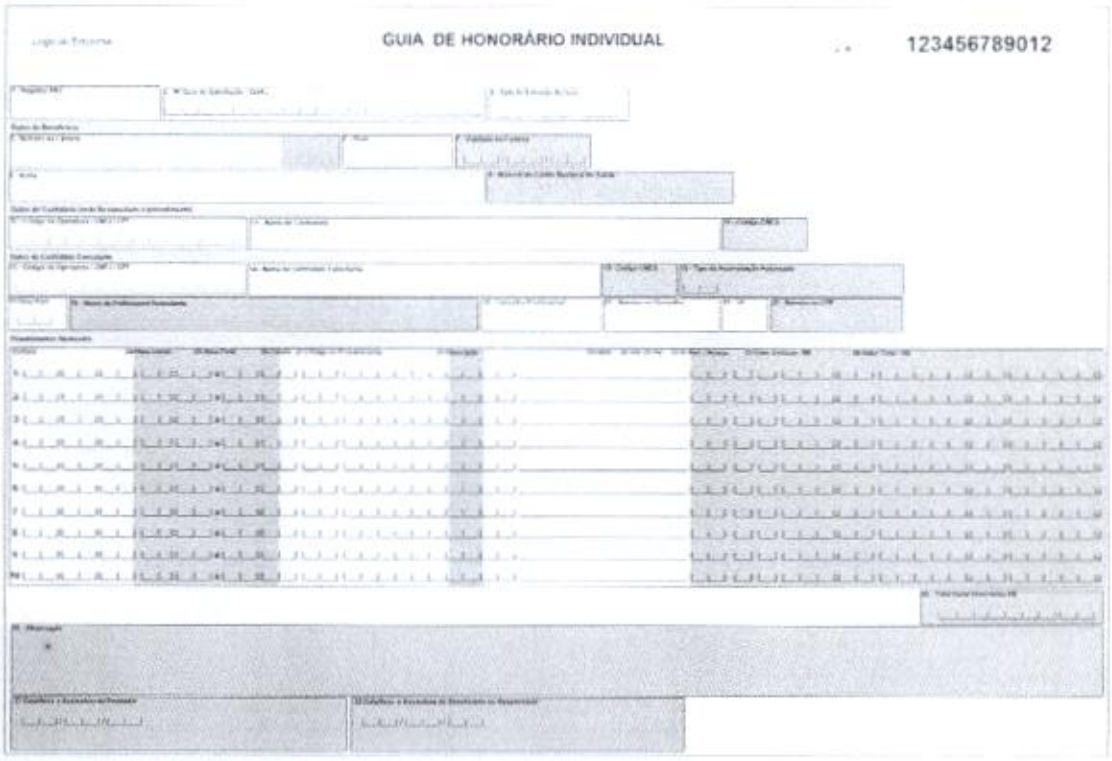
Dados da Autorização

45 - Data Provável da Admissão Hospitalar | 46 - Qtde. Diárias Autorizadas | 47 - Tipo de Acomodação Autorizada

48 - Código na Operadora / CNPJ | 49 - Nome do Prestador Autorizado | 50 - Código CNES

51 - Observação

52 - Data e Assinatura do Médico Solicitante | 53 - Data e Assinatura do Beneficiário ou Responsável | 54 - Data e Assinatura do Responsável pela Autorização



GUIA DE HONORÁRIO INDIVIDUAL

123456789012