

**ANA MARIA PEREIRA
EDUARDO RAFAEL COUTINHO DOS SANTOS**

SISTEMA DE RASTREABILIDADE

Trabalho de conclusão de curso apresentado como requisito parcial à obtenção do grau de Tecnólogo em Informática, Curso de Tecnologia em Informática, Escola Técnica da Universidade Federal do Paraná.

Orientador: Prof. Mario de Paula Soares Filho

Co-Orientador: Prof. Irapuru Haruo Florido

CURITIBA

2005

Pereira, Ana Maria

Sistema de rastreabilidade / Ana Maria Pereira; Eduardo Rafael
Coutinho dos Santos. – Curitiba, 2005.

223 f.: il.

Inclui bibliografia

Orientadores: Mario de Paula Soares Filho

Irapuru Haruo Florido

Monografia (graduação) – Universidade Federal do Paraná, Setor
Escola Técnica.

1. Rastreabilidade 2. Recuperação da informação. I. Santos, Eduardo
Rafael Coutinho dos. II. Soares Filho, Mario de Paula. III. Florido, Irapuru
Haruo. IV. Universidade Federal do Paraná. Setor Escola Técnica. V.
Título.

CDD 20. ed. 004

AGRADECIMENTOS

A todos aqueles que, direta ou indiretamente, contribuíram para a realização deste trabalho.

À

Fabiane Ceruti,

que, mesmo dedicada às suas pesquisas, sempre nos atendeu com atenção e entusiasmo, nos envolvendo e compartilhando seu conhecimento.

Geandré Gomes de Oliveira,

que, sendo muito capacitado e competente no que se refere à conhecimentos técnicos, nos divertiu e alegrou em diversos finais de semana que estudamos juntos durante o projeto.

Nossos pais,

que, mesmo distantes, sempre torceram para que terminássemos logo o projeto, para que tivéssemos mais tempo junto deles. Esperamos que com a conclusão dessas atividades, eles se orgulhem tanto de nós, como nos orgulhamos deles.

Amigos e colegas,

com quem, nessa etapa, entre festas e aulas, aprendemos juntos a aproveitar a vida e assumir responsabilidades. Vocês sempre estarão em nossas mentes e corações.

SUMÁRIO

LISTA DE FIGURAS.....	iv
LISTA DE QUADROS.....	vii
LISTA DE ABREVIATURAS.....	viii
RESUMO.....	ix
INTRODUÇÃO.....	1
JUSTIFICATIVA.....	2
OBJETIVOS.....	3
RASTREABILIDADE.....	4
REVISÃO DE LITERATURA.....	4
ANÁLISE DE REQUISITOS.....	9
DELIMITAÇÃO DO ESCOPO.....	10
LEVANTAMENTO DE TECNOLOGIAS.....	12
LINGUAGEM DE PROGRAMAÇÃO.....	12
PHP 5.....	12
PHP e XML.....	12
ARMAZENAMENTO DOS DADOS.....	14
XML.....	14
WWWIsis.....	14
Tamino - servidor XML Nativo.....	15
TECNOLOGIAS AUXILIARES.....	16
DHTML.....	16
AJAX.....	16
Parser DOM X Parser SAX.....	17
XSL.....	18
XSLT.....	18
DTD.....	18
XQuery.....	19
XForms.....	19
PHP SimpleXML.....	20
FERRAMENTAS UTILIZADAS.....	21
AMBIENTE DE PRODUÇÃO.....	21
REQUISITOS DO SERVIDOR.....	21
REQUISITOS DO CLIENTE.....	21
RESULTADOS.....	22
CONCLUSÃO.....	23
TRABALHOS FUTUROS.....	24

REFERÊNCIAS.....	25
APÊNDICE 1- DEFINIÇÃO DAS ESTRUTURAS XML	28
APÊNDICE 2 - CASOS DE USO	33
APÊNDICE 3 - DESCRIÇÃO RESUMIDA DOS CASOS DE USO	34
APÊNDICE 4 - FLUXO DE EVENTOS DOS CASOS DE USO	37
APÊNDICE 5 - DIAGRAMA DE CLASSES	49
APÊNDICE 6 - DIAGRAMAS DE OBJETOS.....	50
APÊNDICE 6 - DIAGRAMA DE TELAS	51
APÊNDICE 7 – DIAGRAMAS DE SEQUÊNCIA E COLABORAÇÃO.....	52
APÊNDICE 8 - CÓDIGO FONTE	114

LISTA DE FIGURAS

Figura 01 – Estrutura de entidade	10
Figura 02 – Estrutura lógica de registros	11
Figura 03 – Diagrama de casos de uso.....	33
Figura 04 – Diagrama de classes.....	49
Figura 05 – Diagrama de Objetos	50
Figura 06 – Diagrama de telas	51
Figura 07 - Tela cadastro de usuários	52
Figura 08 - Seqüência Incluir Usuário	53
Figura 09 - Seqüência alterar Usuário	54
Figura 10 - Seqüência excluir Usuário	55
Figura 11 - Colaboração Incluir Usuário	55
Figura 12 - Colaboração alterar Usuário	56
Figura 13 - Colaboração excluir Usuário	56
Figura 14 - Tela cadastro de campos	57
Figura 15 - Seqüência Incluir Campo	57
Figura 16 - Seqüência Alterar Campo	58
Figura 17 - Seqüência Excluir Campo	59
Figura 18 - Colaboração Incluir Campo	59
Figura 19 - Colaboração Alterar Campo	60
Figura 20 - Colaboração Excluir Campo	60
Figura 21 - Lista de Entidades	61
Figura 22 - Tela Montar entidade	61
Figura 23 - Seqüência Cadastrar Entidades	62
Figura 24 - Seqüência Alterar Entidade	63
Figura 25 - Seqüência excluir Entidade	64
Figura 26 - Colaboração Cadastrar Entidades	65
Figura 27 - Colaboração Alterar Entidade	65
Figura 28 - Colaboração excluir Entidade	66
Figura 29 - Lista de Origens	67

Figura 30 - Tela montar Origem	67
Figura 31 - Seqüência Incluir Origem	68
Figura 32 - Seqüência alterar Origem	69
Figura 33 - Seqüência Excluir Origem	70
Figura 34 - Colaboração Incluir Origem	71
Figura 35 - Seqüência alterar Origem	72
Figura 36 - Colaboração Excluir Origem	73
Figura 37 - Lista de fases	74
Figura 38 - Tela Montar fase	74
Figura 39 - Seqüência Cadastrar Fases	75
Figura 40 - Seqüência Alterar Fases	76
Figura 41 - Seqüência excluir Fase	77
Figura 42 - Colaboração Cadastrar Fases	78
Figura 43 - Colaboração Alterar Fases	79
Figura 44 - Colaboração excluir Fase	80
Figura 45 - Lista de processos	81
Figura 46 - Tela montar processo	81
Figura 47 - Seqüência cadastrar Processo	82
Figura 48 - Seqüência alterar processo	83
Figura 49 - Seqüência excluir processo	84
Figura 50 - Colaboração cadastrar Processo	85
Figura 51 - Colaboração alterar processo	86
Figura 52 - Colaboração excluir processo	87
Figura 53 - Tela alterar senha	88
Figura 54 - Seqüência Alterar Senha	88
Figura 55 - Colaboração Alterar Senha	89
Figura 56 - Registro de entidade	90
Figura 57 - Tela Agrupar entidades	90
Figura 58 - Seqüência agrupar entidades	91
Figura 59 - Seqüência desagrupar entidade	92

Figura 60 - Colaboração agrupar entidades	93
Figura 61 - Colaboração desagrupar entidade	93
Figura 62 - Tela Incluir Registro de Entidade	94
Figura 63 - Seqüência Incluir Registro de Entidade	95
Figura 64 - Seqüência Alterar Registro de Entidade	96
Figura 65 - Seqüência Excluir Registro de Entidade	97
Figura 66 - Colaboração Incluir Registro de Entidade	98
Figura 67 - Colaboração Alterar Registro de Entidade	98
Figura 68 - Colaboração Excluir Registro de Entidade	99
Figura 69 - Registro de Fase	100
Figura 70 - Tela incluir Registro de Fase	100
Figura 71 - Seqüência Incluir Registro de Fase	101
Figura 72 - Seqüência Alterar Registro de Fase	102
Figura 73 - Colaboração Incluir Registro de Fase	103
Figura 74 - Colaboração Alterar Registro de Fase	103
Figura 75 - Registro de Origem	104
Figura 76 - Incluir Registro de Origem	104
Figura 77 - Seqüência Incluir Registro de Origem	105
Figura 78 - Seqüência Alterar Registro de Origem	106
Figura 79 - Colaboração Incluir Registro de Origem	107
Figura 80 - Colaboração Alterar Registro de Origem	107
Figura 81 - Registro de Processo	108
Figura 82 - Incluir Registro de Processo	108
Figura 83 - Seqüência Incluir Registro de Processo	109
Figura 84 - Seqüência Alterar Registro de Processo	110
Figura 85 - Seqüência Excluir Registro de Processo	111
Figura 86 - Colaboração Incluir Registro de Processo	112
Figura 87 - Colaboração Alterar Registro de Processo	112
Figura 88 - Colaboração Excluir Registro de Processo	113

LISTA DE QUADROS

Quadro 01 – Prioridade nas demandas dos consumidores.....	06
Quadro 02 – Estrutura XML para banco de campos	29
Quadro 03 – Estrutura XML para cadastro de acervos	30
Quadro 04 – Estrutura XML para cadastro de dados	32

LISTA DE ABREVIATURAS

API	-	Aplication Programming Interface
CSS	-	Cascading Style Sheets
DOM	-	Document Object Model
DTD	-	Document Type Definition
HTML	-	HyperText Markup Language
SAX	-	Simple API for XML
SQL	-	Structured Query Language
W3C	-	Word Wide Web <i>Consortium</i>
XML	-	Extensible Markup Language
XSL	-	Extensible Stylesheet Language
XSLT	-	Extensible Stylesheet Language Transformations
AJAX	-	Asynchronous JavaScript and XML
XHTML	-	Extensible HyperText Markup Language

RESUMO

Este trabalho demonstra o processo de desenvolvimento de um sistema de rastreabilidade aplicável à agrícola ou industrial. A rastreabilidade consiste em se armazenar as informações de um produto durante toda a sua cadeia produtiva para garantir a qualidade do produto final. Para facilitar a documentação desta cadeia produtiva organizou-se o sistema de acordo com uma estrutura em que o item principal é chamado de Entidade, cada entidade possui uma Origem e N fases, cada fase possui N processos.

O sistema aqui apresentado foi desenvolvido utilizando-se o conceito de meta dados, em que são armazenados os dados sobre os dados que deverão ser utilizados pelo sistema. Na prática através deste software o administrador do sistema poderá criar todos os campos e telas que devem compor a rastreabilidade do seu produto.

Após a composição das telas (de origem, entidade, fases, processos) o sistema estará pronto para ser utilizado pelo usuário, que fará a inclusão dados, referentes ao processo de produção. O usuário, além de incluir os dados sobre cada entidade, poderá recuperar as informações através de relatórios, que conterão todos os dados registrados para cada entidade, alcançando os objetivos da rastreabilidade.

Para implementar estas funcionalidades no sistema utilizou-se linguagem PHP 5 num servidor Web Apache v.1.3.33. O armazenamento dos dados e dos meta dados é feito em arquivos XML, estes arquivos são manipulados utilizando-se parsers DOM e SAX. Além disso as tecnologias XSL, AJAX, DHTML, Xquery, Xpath também foram pesquisadas e utilizadas na implementação do projeto.

INTRODUÇÃO

No atual cenário econômico brasileiro, o volume dos produtos exportados tem significativo valor na balança comercial. A constante busca pela qualidade é uma premissa máxima para aqueles que estão envolvidos com a produção de gêneros alimentícios. Tal atividade é regida por tendências internacionais, regulada por órgãos ou legislações.

Tendo em vista as exigências de países ou entidades comerciais, os produtores têm procurado mecanismos para um melhor acompanhamento de suas produções. Sendo que a Rastreabilidade tem sido uma nova abordagem neste contexto. Ela busca descrever atividades e relacionar informações com o objetivo de compreender o comportamento dos objetos que são submetidos á ela.

O sistema de Rastreabilidade vem se inserir neste segmento, permitindo que os usuários do sistema possam customizar, conforme as suas necessidades, as interfaces de inserção de dados, para uma eficiente rastreabilidade de dados. Sendo assim, o conceito de metadados, onde são criadas estruturas de dados complexas e flexíveis, é a base deste projeto.

Não sendo uma abordagem convencional, o uso de metadados requer uma alternativa capaz se adequar a este modelo. A linguagem de marcação XML possui características que a enquadram neste formato. Portanto, optou-se pela sua utilização no armazenamento dos metadados.

Sendo assim, o projeto, além de suprir uma crescente necessidade do mercado mundial, busca uma solução através de uma tecnologia pouco explorada, demonstrando assim os resultados positivos diante de uma profunda busca pelo conhecimento e inovação.

JUSTIFICATIVA

A rastreabilidade é um mecanismo que permite identificar a origem do produto desde o campo até o consumidor, podendo ter sido, ou não, transformado ou processado. É um conjunto de medidas que possibilitam controlar e monitorar todas as movimentações nas unidades, de entrada e de saída, objetivando a produção de qualidade e com origem garantida (Pallet *et al.*, 2003).

Neste contexto, um sistema de rastreabilidade poderia apontar um problema de segurança numa região específica da cadeia de produção, por exemplo, em um produtor ou até mesmo numa plantação. E assim, identificando o problema, poder-se-ia retirar do mercado os produtos irregulares e corrigir os problemas exatamente onde eles ocorreram, acelerando assim o processo e diminuindo os impactos econômicos negativos aos participantes da cadeia de produção que não forem responsáveis pelos erros.

Um fator importante, que tornou a rastreabilidade essencial nos dias de hoje, foram as crises alimentares ocorridas na Europa, a partir de 1996, envolvendo segurança alimentar. A partir daí os sistemas de rastreabilidade passaram a ter importância considerável no mercado internacional, deixando clara necessidade de se realizar este trabalho.

OBJETIVOS

Implementar um sistema que permita rastrear todas as etapas da cadeia produtiva de grãos, ou qualquer outro produto, para a verificação de sua qualidade final.

Objetivos específicos:

1. Permitir que o administrador do sistema possa definir toda a estrutura de dados do sistema, criando os campos e as telas que serão utilizadas para criar o histórico dos produtos.
2. Disponibilizar as informações geradas pelo sistema para consultas, tornando possível rastrear o processo produtivo.
3. Permitir o agrupamento e desagrupamento de entidades, especificamente para o caso da produção de grãos. E possibilitar que estes agrupamentos possam ser observados claramente pelo usuário, possibilitando recuperar os dados num histórico não linear.
4. Demonstrar a linguagem XML como ferramenta experimental de armazenamento e recuperação de dados.

RASTREABILIDADE

REVISÃO DE LITERATURA

A rastreabilidade é definida pela Organização Internacional para a Normatização (ISO 8402:1994) como “a habilidade de rastrear a história, aplicação ou localização de uma entidade através de identificação registrada”. A base da rastreabilidade encontra-se no desenvolvimento de sistemas que forneçam informação sobre todo o ciclo de vida dos produtos alimentares, “da fazenda – ou – mar – para o garfo” (FAO, 2003).

O conceito de rastreabilidade difere do conceito de certificação. A rastreabilidade é um processo mecânico que documenta todos os estágios de produção e distribuição pelos quais os alimentos irão passar. Já a certificação é uma afirmação que assegura que certas especificações, por exemplo, a colheita, processamento ou manuseio, foram realizadas em conformidade com padrões ambientais, sociais, de segurança alimentar e de qualidade. Portanto, é importante atentar que os dois conceitos não são sinônimos (FAO, 2003).

A rastreabilidade significa que cada segmento da cadeia alimentar pode seguir o rastro de um alimento e conhecer toda a sua história, antes e depois deste segmento: saber a sua procedência, por onde passou, etc. A rastreabilidade significa, portanto, maior informação e responsabilidade, e exige a aplicação de um sistema eficaz de identificação do produto, desde a sua produção até a sua comercialização (Costa & Euclídes Filho, 2002).

Segundo Schaeffer & Caugant (1998) citado por Lopes (2003), o conceito de rastreabilidade envolve a recomposição da história do produto alimentício. Pode ser útil estabelecer:

- A origem exata de uma produção dos animais ou do vegetal, com os vários fatores que incorporam seu desenvolvimento;
- Histórico dos processos aplicado ao produto;
- A distribuição e a localização do produto final.

De acordo com Joram & Gryna (1992) citado por Souza (2001), a rastreabilidade teria diversas finalidades tais como:

- Assegurar que apenas materiais e componentes de qualidade entrem no produto final;
- Identificar clara e explicitamente produtos que são diferentes, mas que se parecem a ponto de serem confundidos entre si;
- Permitir o retorno de produto suspeito numa base precisa;
- Localizar falhas e tomar medidas corretivas a preço mínimo.

Referências norte-americanas consideram que a rastreabilidade de um atributo particular de um alimento se estabelece quando as características deste atributo são sistematicamente registradas desde a origem do alimento até o seu consumo final. Registrar e transmitir informações sobre produtos alimentares em pontos específicos ao longo da cadeia de distribuição pode ter um número variado de propósitos práticos, incluindo o controle da qualidade do produto ou até mesmo a gestão da cadeia de suprimentos (Mariuzzo & Lobo, 2003).

Esses mesmos autores salientaram que com isso, o mercado norte-americano tenta numerar as motivações do setor privado para a implementação da rastreabilidade, sendo que para eles, os fornecedores de alimentos que atuam no setor privado possuem no mínimo três principais motivos para estabelecer sistemas de rastreabilidade:

- Diferenciar e comercializar produtos alimentares com atributos de qualidade subentendidos ou não detectáveis;
- Facilitar o retorno de informações na cadeia produtiva em relação à segurança alimentar e à qualidade;
- Melhorar o gerenciamento da cadeia de fornecedores e clientes.

Os produtos rastreados possuem um diferencial no mercado, dessa forma tornam-se mais competitivos e menos sujeitos à instabilidades o mundo globalizado. Torna-se mais fácil a detecção de problemas ocorridos durante o processo de produção, bem como a adoção de medidas preventivas e de ação imediata no foco localizado (Pallet *et al.*, 2003).

Machado (2000) definiu a importância da rastreabilidade para os segmentos de distribuição e varejo e da indústria de alimentos:

- É diferencial de competitividade;
- Fortalece a imagem institucional da empresa;
- Auxilia no posicionamento da marca no mercado;
- Estimula a concorrência através da diferenciação da qualidade;
- Estreita a relação com os fornecedores.

Para Costa & Euclides Filho (2002), a rastreabilidade traz consigo a garantia de transparência ao consumidor do conteúdo/origem do produto que lhe é ofertado para a compra, mediante uma rotulagem precisa, especialmente onde o produto final tem características que não podem ser prontamente testadas. A rastreabilidade é essencial para controlar e garantir a qualidade em estágios particulares da cadeia alimentar, ou seja, em etapas específicas do processo produtivo.

Desde os anos 70, as necessidades dos consumidores têm mudado no mundo inteiro e as comunicações tornaram realidade a interação entre as culturas. Isso levou a uma certa padronização nas demandas e o consumidor está hoje ciente de suas necessidades, exigindo alimentos de melhor qualidade. O Quadro 1 indica o aumento nas prioridades dos consumidores (Nääs, 2001). O aumento de viagens internacionais tem favorecido, em várias partes do mundo, o surgimento da chamada “alimentação étnica”. Estes são apenas alguns exemplos de como os hábitos de consumo alimentar vêm se modificando em praticamente todo o mundo.

Quadro 1: Prioridade nas demandas dos consumidores.

1970	1980	1990	2000	2002
Preço	Frescor Qualidade Variedade	Bem estar dos Animais e trabalhadores Segurança Reciclagem	Tradição Etnia Sustentabilidade	Ética Rastreabilidade

Fonte: Holroyd (2000), citado por Nääs (2001).

De acordo com Holroyd (2000), o futuro do comércio de alimentos depende principalmente da indústria reagir com respostas eficientes aos seguintes conceitos: honestidade, abertura, informações detalhadas, garantia de qualidade, rastreabilidade e flexibilidade de adoção de novas técnicas. Para o comércio varejista ou as distribuidoras somente é possível construir um bom negócio quando a garantia de qualidade é constante, e esta garantia somente é possível com a aplicação da rastreabilidade.

De acordo com Nääs (2003), a rastreabilidade é um processo crescente e irreversível, impulsionado pelas economias de escala, decorrentes dos avanços tecnológicos e da demanda do mercado importador que, cada vez mais, exige ética e transparência nos processos de produção e distribuição dos produtos.

O setor agroindustrial brasileiro, também precisa se conscientizar de que a segurança alimentar é um assunto que doravante vai ser uma condicionante obrigatória para competir no mercado e que cada vez mais será preciso dar transparência aos procedimentos de produção e industrialização para que os consumidores possam se certificar da qualidade dos alimentos que consomem (Altmann, 2001).

Outro aspecto que passa a ser considerado quando se fala em agronegócio é a visão de Cadeia Produtiva que pressupõe que as empresas não podem mais atuar sozinhas, devendo considerar a competitividade de seus fornecedores, compradores e de todos os agentes participantes do encadeamento de atividades, como forma de sustentar a sua própria competitividade e manter um posicionamento sustentável. Começa a surgir a cooperação entre os mais diversos integrantes da cadeia, que atuam de forma coordenada e competem com outras cadeias, como se fossem uma única empresa, estabelecendo estratégias e distribuindo as vantagens conquistadas por todos os integrantes do sistema (Mariuzzo & Lobo, 2003).

Este cenário apresenta novos desafios às organizações envolvidas na Cadeia Produtiva de Alimentos, pois estas deverão de forma integrada, fazer uso mais eficiente dos seus insumos, desenvolvendo processos e produtos mais limpos, gerenciando os recursos naturais e humanos de forma mais responsável e garantir a

segurança alimentar do produto final, práticas estas que se tornam viáveis a partir da aplicação das Boas Práticas Agrícolas (Mariuzzo & Lobo, 2003).

ANÁLISE DE REQUISITOS

Conforme as informações levantadas, para se rastrear um produto deve-se cadastrar as entidades ou lotes a serem rastreados e durante toda a cadeia produtiva armazenar informações sobre todos os procedimentos que são realizados no produto, por exemplo, na produção de grãos seriam armazenados os dados sobre adubação, aplicação de herbicidas, secagem etc.

Também se podem colher informações do ambiente em que ocorreu a produção, como tipo de solo, temperatura média, quantidade de chuvas e outros. Os dados vão sendo colhidos desde o início da produção passando por todas as fases até que seja vendido ao consumidor final.

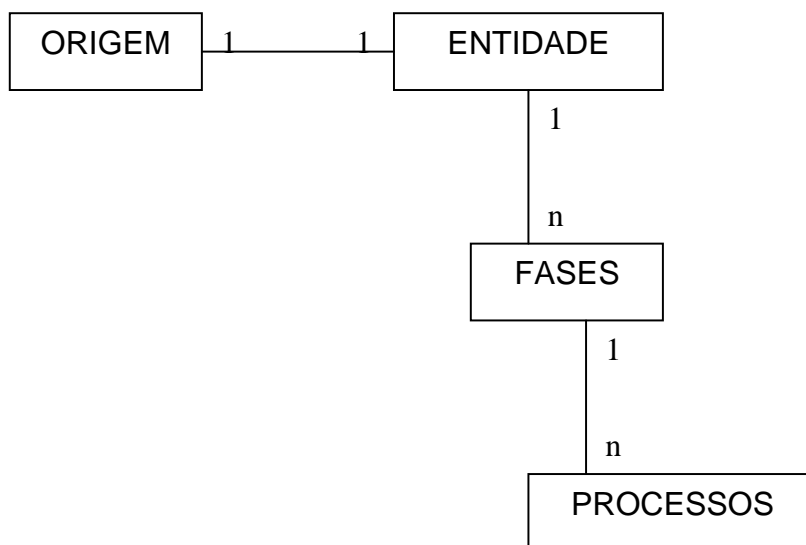
Para isso é necessário criar uma estrutura em que se registrem os dados da Origem do produto e os dados do próprio produto, ou entidade, que será rastreado. Após o cadastramento destas informações, o sistema deve ser alimentado pelo operador a cada fase e processo que ocorre na produção.

DELIMITAÇÃO DO ESCOPO

De acordo com as informações apresentadas no capítulo anterior o sistema foi organizado com o seguinte conceito: cada Indivíduo ou lote de indivíduos a ser rastreados é chamados de “Entidade”, toda a entidade pode possuir uma “Origem”, que diz de onde ela veio. As etapas da cadeia produtivas são chamadas “Fases” e durante cada fase vão ocorrer processos que devem ser registrados para compor a rastreabilidade de cada entidade.

A Figura 01 tenta ilustrar a estrutura básica definida para a rastreabilidade.

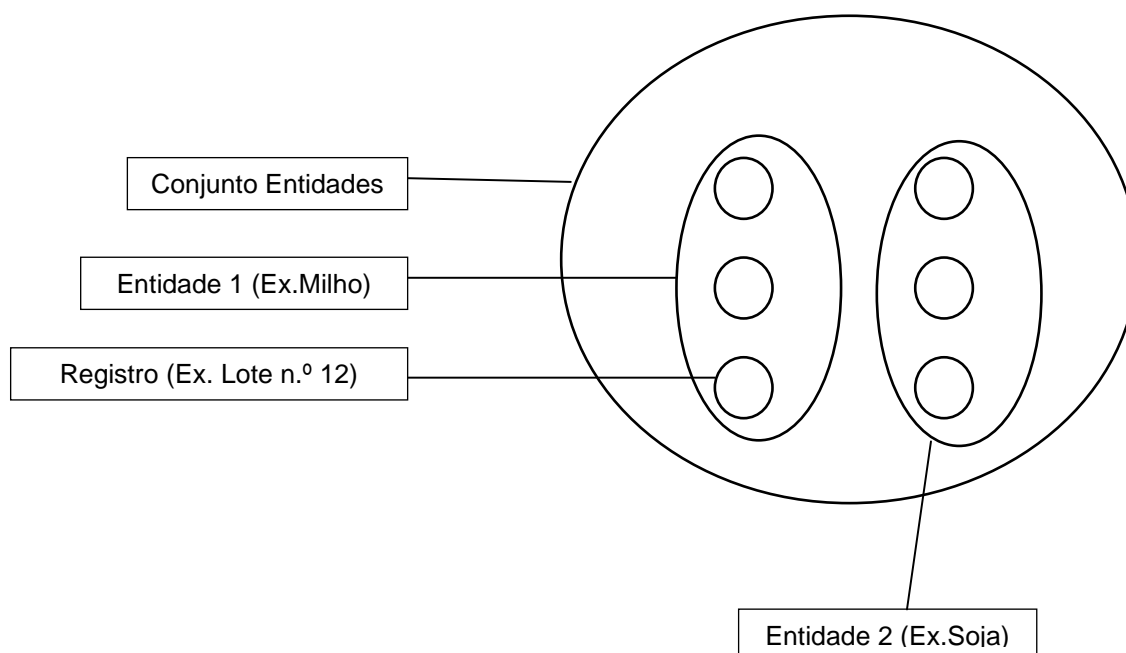
Figura 01 – Estrutura de entidade



Para a implementação optou-se por utilizar meta dados, portanto, para que seja possível fazer a implementação nesse modelo, o administrador do sistema deverá cadastrar os campos e as telas que pertencem ao sistema. Somente depois de definir as telas de Origem, Entidades, Fases e Processo é que será possível inserir efetivamente os dados da rastreabilidade.

Quando as telas estiverem cadastradas, o operador do sistema poderá iniciar a inclusão dos registros. O Conjunto de Entidades contém as estruturas que são definidas pelo administrador, sendo que o operador selecionará uma para incluir registros. Desta maneira cada tipo de Entidade poderá ter diversos registros conforme a Figura 2.

Figura 2 – Estrutura lógica de registros



Conforme as informações contidas neste capítulo alguns termos foram utilizados para denominar elementos do sistema. A seguir seguem suas definições no contexto do sistema:

- Estrutura de entidade: é a definição do perfil de um tipo de objeto que se deseja rastrear, contendo elementos que descrevem as características deste grupo.
- Entidade: é o objeto da rastreabilidade, ou seja, o produto a ser rastreado pelo sistema.
- Origem: refere-se à procedência do produto, ou às informações que se tem sobre ele antes de iniciar a rastreabilidade.
- Fases: são as etapas do ciclo de produção.
- Processo: são as atividades às quais o produto é submetido durante a produção.
- Registros: são as ocorrências de elementos de um tipo de Entidade.

LEVANTAMENTO DE TECNOLOGIAS

Para a realização deste trabalho foi necessário fazer uma exaustiva prospecção tecnológica, buscando conhecer as tecnologias existentes no mercado e selecionar aquelas que fossem mais interessantes para a execução do projeto. Durante um período de três meses foram estudadas diversas tecnologias. Segue abaixo um pequeno resumo das suas principais características, citando os motivos que motivaram a sua utilização ou não no projeto.

LINGUAGEM DE PROGRAMAÇÃO

PHP 5

É uma linguagem de programação de interpretada, utilizada para gerar páginas dinâmicas na Web. Pode ser embutida no código HTML e, além disso, destaca-se pela extrema facilidade com que lida com servidores de base de dados. A sintaxe é bastante simples e fácil de aprender, além disso, possui suporte a Orientação a objetos e ao XML, por esta razão foi escolhida para o desenvolvimento deste projeto.

PHP e XML

Na linguagem PHP as funções de manipulação estão presentes desde sua origem. Mais recentemente, como a especificação XML tornou-se proeminente como linguagem de troca de dados, o PHP incorporou funções que o tornaram uma ferramenta fácil para trabalhar com documentos XML. Devido à natureza e o formato dos documentos XML a maior parte do trabalho para criar funções XML para o PHP esteve focado em analisar e manipular os documentos XML mantendo-se em conformidade com a especificação do W3C. Para efetivamente analisar e manipular os

documentos XML estas funções têm que permitir trabalhar com os nomes e valores dos elementos e dos atributos bem como com os muitos outros tipos de componentes dos documentos XML. No manual do PHP são apresentadas funções de parser DOM, de parser SAX, SimpleXML, XML-RP (experimental) e funções de processador de XSL/XSLT.

ARMAZENAMENTO DOS DADOS

XML

O XML é uma linguagem de marcação e é também a base de documentos estruturados que representam não somente a informação a ser passada, mas também os metadados, que definem o seu significado e a estrutura da informação.

Ele é uma maneira simples de delimitar os dados do texto. As *tags* do XML dão nome ao conceito descrito, e atributos identificados modificam as estruturas rotuladas. Desta forma, os dados em XML são auto-descritivos, ou seja, fáceis de compreender e manipular.

Apesar de tudo o XML também possui as suas desvantagens: não possui mecanismos eficientes de armazenamento que substituam os existentes em bancos de dados relacionais como garantir as restrições de integridade, manipulação de dados, armazenamento e recuperação de dados fisicamente e otimizações de consulta. E os seus documentos podem ser de 3 a 20 vezes maiores do que os representados em formato binário e texto.

Resultado: Apesar das desvantagens citadas, escolhemos utilizar o XML para a implementação do repositório, pois é uma técnica poderosa de estruturar os dados, trocá-los e apresentá-los independentemente da plataforma.

WWWIsis

O WWWISIS é o servidor de Internet para bases de dados Isis. Opera via CGI (Common Gateway Interface) do Hypertext Transport Protocol (HTTP) e, a partir da versão 4.0, conhecida também como WXIS, utiliza uma linguagem de script baseada em XML, o IsisScript.

A principal característica do IsisScript é a sua capacidade de processamento de entidades Isis por meio da transferência de dados de um navegador WWW para o processo CGI.

Com o WWWISIS, estão disponíveis linguagem de pesquisa, formatação e atualização de registros, sendo por isso possível a montagem de formulários de pesquisa, bem como a implementação de operações para a atualização de bases de dados.

A saída do IsisScript para o navegador WWW pode ser gerada via linguagem de formatação Isis. A possibilidade de incluir códigos HTML nas especificações do formato permite desenvolver uma interface de pesquisa totalmente gráfica, em ambiente de Internet ou Intranet.

Tamino - servidor XML Nativo

O Servidor Tamino é uma solução para o armazenamento, a troca e a recuperação de documentos XML. Por armazenar XML em seu formato nativo, o Tamino oferece desempenho e escalabilidade muito bons para trabalhar com dados semiestruturados.

A vantagem do Tamino é sua arquitetura baseada em XML que oferece uma solução de muito boa performance. Ele utiliza o próprio XML para estruturar, organizar e armazenar as informações. E existem muitas vantagens em se armazenar o XML de forma nativa.

Resultado: apesar de ser a solução ideal para armazenamento dos dados este é um banco pago, o que inviabiliza sua utilização.

TECNOLOGIAS AUXILIARES

DHTML

É um conjunto de tecnologias usadas para criar páginas Web dinâmicas e interativas. É uma combinação do HTML, folhas de estilo (CSS), linguagens de script (JavaScript), e o DOM.

DHTML não é uma recomendação do W3C, é um termo criado para designar o uso combinado das tecnologias citadas acima.

Com DHTML o desenvolvedor web pode manipular os atributos de formatação e posicionamento de elementos HTML que são apresentados pelo browser. Fazendo, desta forma, com que elementos sejam adicionados ou removidos da tela dinamicamente ou que os atributos de formatação mudem devido à ações como clicar, passar o ponteiro sobre um objeto, etc.

Resultado: O javascript juntamente com as técnicas de DHTML foram escolhidos para realizar algumas funções no sistema, em que não se faz necessário fazer uma requisição ao servidor, mas apenas acessar e alterar propriedades do HTML tornando as páginas mais dinâmicas.

AJAX

É um conjunto de tecnologias que servem para criar aplicações dinâmicas para Internet. Ele é baseado em XHTML e CSS para conteúdo e apresentação, DOM (Document Object Model) para apresentação dinâmica e interação, XMLHttpRequest para recuperação assíncrona de dados, XML e javascript para unir tudo. Ele é a peça chave por traz do Gmail.

Essa técnica consiste em utilizar Javascript, XML e XMLHttpRequest de forma a criar páginas que executem código sem serem recarregadas, ou seja, sem atualizar as páginas cada vez que uma requisição é feita ao servidor.

Resultado: A técnica foi utilizada em alterações dinâmicas na tela nos casos em que é necessário fazer uma requisição ao servidor.

Parser DOM X Parser SAX

SAX é uma API que permite acessar rapidamente informações em um arquivo XML, mas não permite a sua edição, já o DOM permite um acesso completo ao arquivo.

Dom é bastante eficaz quando se necessita obter muitas informações contidas em um documento XML, permitindo que qualquer parte do documento seja acessada a qualquer momento.

O forte do SAX é sua eficiência. Se for necessário apenas recuperar uma pequena parte das informações de um documento XML e sabe-se onde elas estão, é fácil escrever um código que recupere estas informações. Com documentos grandes é fácil ver como este parser é superior. Ao contrário do DOM, que cria uma estrutura de árvore de nodos para todo o documento, SAX permite que se gaste tempo somente com aquilo que é necessário.

É interessante notar que muitas implementações DOM usam SAX como seu parser sublinhando, criando árvores de nodos a partir de eventos do SAX, retornando os dados para a função que o chamou.

O Parser DOM é mais recomendado para documentos pequenos em que se necessita recuperar uma grande quantidade de informações. O Parser SAX oferece uma performance muito melhor com documentos grandes, quando se necessita recuperar uma pequena parte das informações contidas em um documento.

Resultado: foi escolhido o DOM que é mais útil, por permitir manipular os arquivos XML alterando seu conteúdo.

XSL

XSL é uma linguagem de definição de folhas de estilo, semelhante ao CSS, só que bem mais poderosa e complexa. É composta por XSLT, responsável por transformar o documento original e FO (Formating Objects), responsável por aplicar a formatação ao arquivo transformado.

XSLT

XSLT é a parte mais importante do XSL. É usado para transformar um documento XML em um outro tipo de documento que seja reconhecido pelo navegador, como HTML e XHTML. Normalmente o XSLT transforma cada elemento XML em um elemento HTML. Com o XSLT é possível adicionar e remover atributos do arquivo de saída. Também é possível reorganizar e ordenar elementos, permitindo escolher quais elementos devem ser mostrados e quais não.

Resultado: utilizamos XSL para gerar diversos relatórios e inclusive a interação das telas de cadastro dos dados.

DTD

O propósito de um DTD é modelar e validar os dados de um documento XML. Ele define a estrutura do documento, assim com especifica uma lista de elementos válidos e seu possível conteúdo. Utilizando o DTD é possível determinar se um arquivo é valido e se segue as normas definidas para sua utilização.

Resultado: não utilizado, pois não foi considerado necessário.

XQuery

É uma a linguagem para buscar e extrair dados de elementos e atributos em arquivos XML baseada em expressões de XPath.

É compatível com a maioria das recomendações W3C, como XML, Namespaces, XSLT, XPath e XML Schema.

Resultado: utilizado na gravação de dados juntamente com XPath.

XForms

Atualmente, os forms são uma parte muito importante da maioria das aplicações web. Um form HTML torna isso possível para aplicações web, para a entrada de dados do usuário.

Hoje, dez anos depois do surgimento dos forms HTML, os usuários de sistemas web realizam transações complexas que inicialmente excediam as limitações dos forms HTML.

Xforms provêm uma forma mais segura e independentemente de manipular entradas de dados. Eles usam XML para definição de dados e HTML ou XHTML para apresentar os dados. Xforms separam os dados lógicos de um form de sua apresentação. Desta maneira os dados dos XForms podem ser definidos independentemente de como o usuário final irá interagir com a aplicação.

Com XForms a descrição e validação de dados está em XML. Os dados são apresentados em um formulário definido em um documento XML e os dados submetidos ao form são transportados pela internet usando XML.

Resultado: Não utilizado por falta de suporte dos navegadores.

PHP SimpleXML

A extensão SimpleXML do PHP oferece funções para converter o XML em um objeto que pode ser processado com propriedades normais de seleção e interações de Array. O suporte ao SimpleXML no PHP5 é instalado automaticamente não é necessário incluir extensões adicionais.

SimpleXML inclui funções para trabalhar com documentos XML que realizam operações comuns como converter uma string em um documento XML formatado e apresentá-lo. A primeira vantagem é que o documento XML transforma-se em um objeto que pode ser processado como outros objetos em PHP com elementos e atributos e seus dados são acessados usando operações normais de objetos. A extensão inclui funções para trabalhar com documentos XML

Resultado: Usado nos métodos construtores de objetos para obter os dados dentro dos arquivos XML.

FERRAMENTAS UTILIZADAS

Ferramenta de modelagem UML: Jude Community 1.4.1 e Java Jdk1.5.0_06

Editor HTML: Macromedia Dreamweaver MX 2004

Editor PHP: PHP Editor

Servidor: WAMP5 com Apache 2.0.55 e PHP 5

AMBIENTE DE PRODUÇÃO

REQUISITOS DO SERVIDOR

Software:

Servidor: Apache 2.0.55 e PHP 5

Sistema Operacional: Windows98 ou superior

Hardware: Processador 1Ghz ou superior, 256 MB de memória.

REQUISITOS DO CLIENTE

Software:

Navegador: Internet Explorer 5.0 ou versão superior

Sistema Operacional: Windows98 ou superior

Hardware: Processador 500 MHz ou superior, 128 MB de memória.

RESULTADOS

O sistema foi implementado de maneira que a criação dos formulários de cadastro seja customizável, atendendo a proposta do projeto quanto à estruturação da rastreabilidade. Existe um mecanismo eficiente de consulta dos dados cadastrados e também é possível fazer agrupamento e desagrupamento das entidades.

De um modo geral, o sistema atingiu os objetivos definidos no início do projeto, com ressalva em relação ao armazenamento e recuperação dos dados, onde, apesar do XML demonstrar a capacidade de desempenhar um papel de repositório de dados, apresenta limitações no que se refere ao armazenamento de dados.

Além disso a equipe que desenvolveu o projeto observou uma considerável evolução no que se refere ao seu conhecimento de tecnologias para desenvolvimento de aplicações baseadas na WEB. Portanto os esforços em estudar tecnologias alternativas se demonstraram gratificantes.

CONCLUSÃO

Conforme foi se desenvolvendo o projeto de rastreabilidade, houve uma real necessidade de se compreender certos métodos de organização de dados, afim de que eles gerem informação e se relacionem de maneira que um registro seja uma representação fiel do que efetivamente acontece.

Por se tratar de dados encadeados em estruturas complexas, o conceito de metadado foi exaustivamente estudado. A ponto de definir uma visão ambiciosa que é possível descrever qualquer estrutura lógica nesse formato.

A linguagem de marcação XML veio atender as necessidades quanto à persistência dos dados. Das estruturas de telas, aos dados cadastrados a partir delas, a rastreabilidade se baseou na definição do metadado, usando como meio arquivos XML.

Num ambiente onde dados são classicamente baseados em bancos relacionais, a utilização de bancos textuais apresentou-se como uma alternativa eficiente para enfrentar problemas não convencionais, como estruturas que não sejam fixas, podendo ajustar-se conforme são alimentadas com dados.

Contudo, para se extrair o máximo do recurso de bancos desse tipo é necessária que o sistema esteja utilizando um gerenciador de banco de dados XML nativo. Garantindo segurança e concorrência independentes da aplicação.

TRABALHOS FUTUROS

Visando suprir as já mencionadas limitações quanto á segurança e concorrência de dados, seria interessante estruturar-se a persistência e recuperação dos dados numa estrutura baseada em um gerenciador de dados XML nativo.

Considerando que o sistema de rastreabilidade mantém uma estrutura de criação genérica permitindo diversos tipos de entidade, a implementação de estruturas padrões, seguindo especificações legislativas e normativas, nacionais ou internacionais, seria um diferencial quanto aos demais sistemas já existentes, com a diferença de poder-se customizar quanto a realidade do ambiente onde se fará a rastreabilidade.

O sistema tem o objetivo específico que é a rastreabilidade, mas permite que dentro da estrutura seja criada uma nova implementação que analise o histórico possibilitando a tomada de decisão, através de emissão de relatórios estatísticos.

REFERÊNCIAS

- AJAX y ASP. Ejemplo de simple de uso.** Disponível em:
<<http://www.soloasp.com.ar/vereje.asp?eje=56>> Acesso em 11 Nov. 2005
- ALTMANN, R. 2001. **Segurança alimentar.** Disponível em:
<<http://www.an.com.br/2001/fev/15/0opi.htm>> Acesso em: 21 Maio 2005
- Berkeley DB XML** Disponível em:
<<http://www.sleepycat.com/products/bdbxml.html>> Acesso em: 09 Dez. 2005
- BORDIN, R. C.; AMSTRONG, G. C. **Projeto Acervo Digital.** Curitiba, 2005. 241 f. Monografia (Graduação em Tecnologia em Informática), Escola Técnica, Universidade Federal do Paraná
- CARVALHO JUNIOR, J. **Definição de um padrão baseado em XML para descrição de normas jurídicas brasileiras.** Curitiba, 2004. 79 f. Dissertação (Mestrado em Informática) – Setor de Ciências Exatas, Universidade Federal do Paraná.
- COSTA, C. N., EUCLIDES F., K. **Identificação Animal e Rastreamento da Produção de Bovinos de Corte e de Leite.** Agrosoft 2002.
- DHTML Tutorial.** Disponível em:
<<http://www.w3schools.com/dhtml/default.asp>> Acesso em: 10 Jul. 2005
- DUNFORD, M. **DOM XML: An Alternative to Expath.** Disponível em:
<<http://www.phpbuilder.com/columns/matt20001228.php3>> Acesso em: 06 Nov 2005
- FADEL, N. S. **Estudo da adequação do uso de XML para o armazenamento de dados históricos.** Curitiba, 2004. 70 f. Dissertação (Mestrado em Informática) – Setor de Ciências Exatas, Universidade Federal do Paraná.
- FAO – Food and Agriculture Organization of the United Nations – **Item 8 of the provisional agenda.** Disponível em:
<<http://www.fao.org/docrep/meeting/004/y3015E.htm>> Acesso em: 21 Maio 2005
- FERREIRA, E. **Ajax para quem só ouviu falar** Disponível em:
<<http://www.tableless.com.br>> Acesso em 05 Nov. 2005
- FERREIRA, E. **Ajax (parte 2): encarando o mundo real** Disponível em:
<<http://www.tableless.com.br>> Acesso em 14 Nov. 2005
- FORD, P. **Berkeley DB XML: An Embedded XML Database** Disponível em:
<<http://www.xml.com/pub/a/2003/05/07/bdb.html>> Acesso em: 09 Dez. 2005

HOLROYD, P. **Tendências do mercado de carne para o novo milênio**. In: CONFERÊNCIA APINCO, 2001, Campinas. **Anais..** Campinas, 2000. p. 93-109.

ISRAELIAN, E. et al. **Uma Introdução às Normas da Série ISO 9000**
Disponível em: < <http://allchemistry.iq.usp.br/pub/metabolizando/bd6c001z.doc> >
Acesso em: 15 Fev. 06

JavaScript Tutorial. Disponível em:
<<http://www.w3schools.com/js/default.asp>> Acesso em: 08 Jul. 2005

JURAM, J. M.; GRINA, F. M. **Controle da qualidade – ciclo dos produtos: do projeto à fabricação**. São Paulo: Makron books, 1992. V. 2.

LOPES, E. **A futura norma de rastreabilidade ISSO 22005** Disponível em:
www.fooddesign.com.br/arquivos/academia/08%20Ellen%20Rastreabilidade%2010-08-05.pdf. Acesso em: 23 Fev. 2006

LOPES, M. A. **Rastreabilidade na bovinocultura**. Lavras: FAEPE/PROEX, 2003.

MACHADO, R. T. M. **Rastreabilidade, tecnologia de informação e coordenação de sistemas agro-industriais**. São Paulo, 2000. 239 f. Tese (Doutorado em administração) Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo .

Manual do PHP - XXVII DOM Functions Disponível em:
<http://br2.php.net/manual/pt_BR/ref.dom.php> Acesso em: 11 Nov. 2005

MARIUZZO, D.; LOBO, D. **Rastreabilidade e segurança alimentar: exigências do mercado consumidor – o caso da fruticultura**. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E À AGROINDÚSTRIA, 4, 2003. Porto Seguro. **Anais ...** , Porto Seguro, 2003.

MUTO, C. A **PHP e MYSQL: guia avançado**. Rio de Janeiro: Brasport, 2004

NÄÄS, I.A. **Rastreabilidade e certificação de suínos no Brasil**. II Conferencia Internacional Virtual sobre Qualidade de Carne Suína, 2001.

NÄÄS, I. A. **A importância da rastreabilidade nas cadeias produtivas de suínos e aves**. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E À AGROINDÚSTRIA, 4, 2003. Porto Seguro. **Anais ...** , Porto Seguro, 2003.

NIEDERAUER, J. **PHP com XML**. 2. e. São Paulo: Novatec, 2002

NIEDERAUER, J. **Desenvolvendo Websites com PHP**. São Paulo: Novatec, 2004

PALLET, D. et al. **Um panorama da rastreabilidade dos produtos agropecuários do Brasil destinados à exportação – carnes, soja e frutas.** Escola Superior de Agricultura "Luiz de Queiroz", Universidade de São Paulo, 2003. Disponível em: <<http://www.cendotec.org.br/prosper/publicacoes/8-rastreabilidade.pdf>> Acesso em: 01 Jan. 2006

RAMALHO, J. **XML: teoria e prática.** São Paulo: Beckerley, 2002

SAES, S. M.; CUNHA, G. J. **Introdução à rastreabilidade .** Disponível em: <<http://www.pcs.usp.br/~laa/html/rastreabilidade2005/gilberto.pdf>> Acesso em: 16 Fev. 2006

SCHAEFFER, E.; CAUGANT, M. **Traçabilité guide pratique pour l'agriculture e l'industrie alimentaire.** Paris, France. ACTA – ACTIA, 1998.

SOUZA, E. L. L. **Preservação de identidade de grãos e a coordenação dos sistemas agro-industriais.** Piracicaba, 2001. Tese (Doutorado em Agronomia), Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo.

Tamino XML Server Disponível em: <<http://www.xml.com/pub/p/119>> Acesso em: 06 Dez. 2005

TRAMONTIN JUNIOR , R. J. **Configuração e Integração de Dados em Plataformas Para Empresas Virtuais.** Florianópolis, 2004. 209 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Santa Catarina

XML Tutorial. Disponível em: <www.w3schools.com/xml/default.asp> Acesso em: 15 Set. 2005

XSLT Tutorial. Disponível em: <<http://www.w3schools.com/xsl/default.asp>> Acesso em: 05 Nov. 2005

XML DOM Tutorial. Disponível em: <<http://www.w3schools.com/dom/default.asp>> Acesso em: 12 Out. 2005

APÊNDICE 1- DEFINIÇÃO DAS ESTRUTURAS XML

Para que o sistema utilize XML como meio armazenamento e organização dos dados é necessário criar uma estrutura padrão para os documentos.

Um dos requisitos básicos para a estrutura dos arquivos é que eles ofereçam a máxima flexibilidade ao sistema, para que o usuário possa definir os campos e as telas do sistema como considerar mais adequado e ao mesmo tempo esta estrutura deve permitir ao sistema administrar e manipular todos os dados encontrados nestes arquivos. Ou seja, o sistema deve possuir uma estrutura fixa e que ofereça a maleabilidade dos dados.

Buscando manter estas características definimos as estruturas hierárquicas dos três principais repositórios de dados do sistema.

São eles:

- Cadastro de campos: deverá conter as definições de todos os campos cadastrados.
- Cadastro de telas: deverá conter as definições das telas cadastradas no sistema e os respectivos campos.
- Cadastro de entidades: deverá conter os dados das entidades cadastradas no sistema. Este é o principal arquivo do sistema pois contém as informações das entidades que devem ser rastreadas. Ele segue a mesma estrutura hierárquica demonstrada na definição conceitual do sistema.

CADASTRO DE CAMPOS

O arquivo que contém o cadastro de campos possui N registros de campo. Cada registro está dentro da tag <campo> e possui um atributo 'CODIGO' que possui um valor único de identificação do registro. O cadastro de cada campo tem os seguintes itens: nome, tipo de dado, tipo de HTML e valores. Desta forma a estrutura do XML é apresentada no Quadro 2.

Quadro 2 : Estrutura XML para banco de campos

```
<banco_campos>
  <campo CODIGO="1" ATIVO="S">
    <nome>Talhão</nome>
    <tphtml>text</tphtml>
    <valores>
      <opcao/>
    </valores>
  </campo>
  <campo CODIGO="2" ATIVO="S">
    <nome>Quantidade</nome>
    <tphtml>text</tphtml>
    <valores>
      <opcao/>
    </valores>
  </campo>
  <campo CODIGO="3" ATIVO="S">
    <nome>Semente</nome>
    <tphtml>text</tphtml>
    <valores>
      <opcao>0,00</opcao>
    </valores>
  </campo>
</banco_campos>
```

CADASTRO DE ACERVOS

O arquivo que contém o cadastro de Acervos possui N registros de tela. Cada registro está dentro da tag correspondente ao tipo de acervo (entidade, fase, processo ou origem) e possui os atributos 'CODIGO', que possui um valor único de identificação do registro, 'tipo' e 'nome'. Os registros de cada acervo contêm N campos, estes registros de campos seguirão a mesma estrutura mostrada no Cadastro de campos. A estrutura XML resultante é apresentada no Quadro 3.

Quadro 3 : Estrutura XML para cadastro de acervos

```
<banco_telas>
  <ENTIDADE CODIGO="1" ATIVO="S" tipo="Entidade" nome="Milho">
    <campo CODIGO="11" ATIVO="S">
      <nome>Peso</nome>
      <tphtml>text</tphtml>
      <valores><opcao/></valores>
    </campo>
    <campo CODIGO="46" ATIVO="S">
      <nome>Variedade</nome>
      <tphtml>text</tphtml>
      <valores><opcao/></valores>
    </campo>
    <Origem>1</Origem>
    <Fase>3</Fase>
    <Fase>2</Fase>
    <Fase>1</Fase>
  </ENTIDADE>
  <PROCESSO CODIGO="2" ATIVO="S" tipo="Processo" nome="Termometria">
    <campo CODIGO="25" ATIVO="S">
      <nome>Data</nome>
      <tphtml>text</tphtml>
      <valores><opcao/></valores>
    </campo>
    <campo CODIGO="38" ATIVO="S">
      <nome>Temperatura média</nome>
      <tphtml>text</tphtml>
      <valores><opcao/></valores>
    </campo>
    <campo CODIGO="39" ATIVO="S">
      <nome>Temperatura ambiente</nome>
      <tphtml>text</tphtml>
      <valores><opcao/></valores>
    </campo>
  </PROCESSO>
</banco_telas>
```

CADASTRO DE DADOS

O cadastro de dados possui N registros de Entidade. O registro de cada entidade esta na tag <ENTIDADE>, que possui os atributos 'ID', 'CODIGO' e 'ATIVO'.

As entidades possuem os seguintes dados:

'Campo', pode ser repetido diversas vezes e possui a propriedade NOME, estes dados representam os campos da tela de entidade;

'Origem', é um dado único, representa os dados da origem desta entidade de acordo com a tela de origem e pode possuir N campos;

'Fases', possui N dados Fase, que devem possuir N dados Campo, de acordo com a tela de fases e um dado Processos. Finalmente Processos possui N dados Processo, que irão conter N Campos com os dados do processo, conforme o tipo de processo/tela usado. O XML resultante é mostrado no Quadro 4.

Quadro 4 : Estrutura XML para cadastro de dados

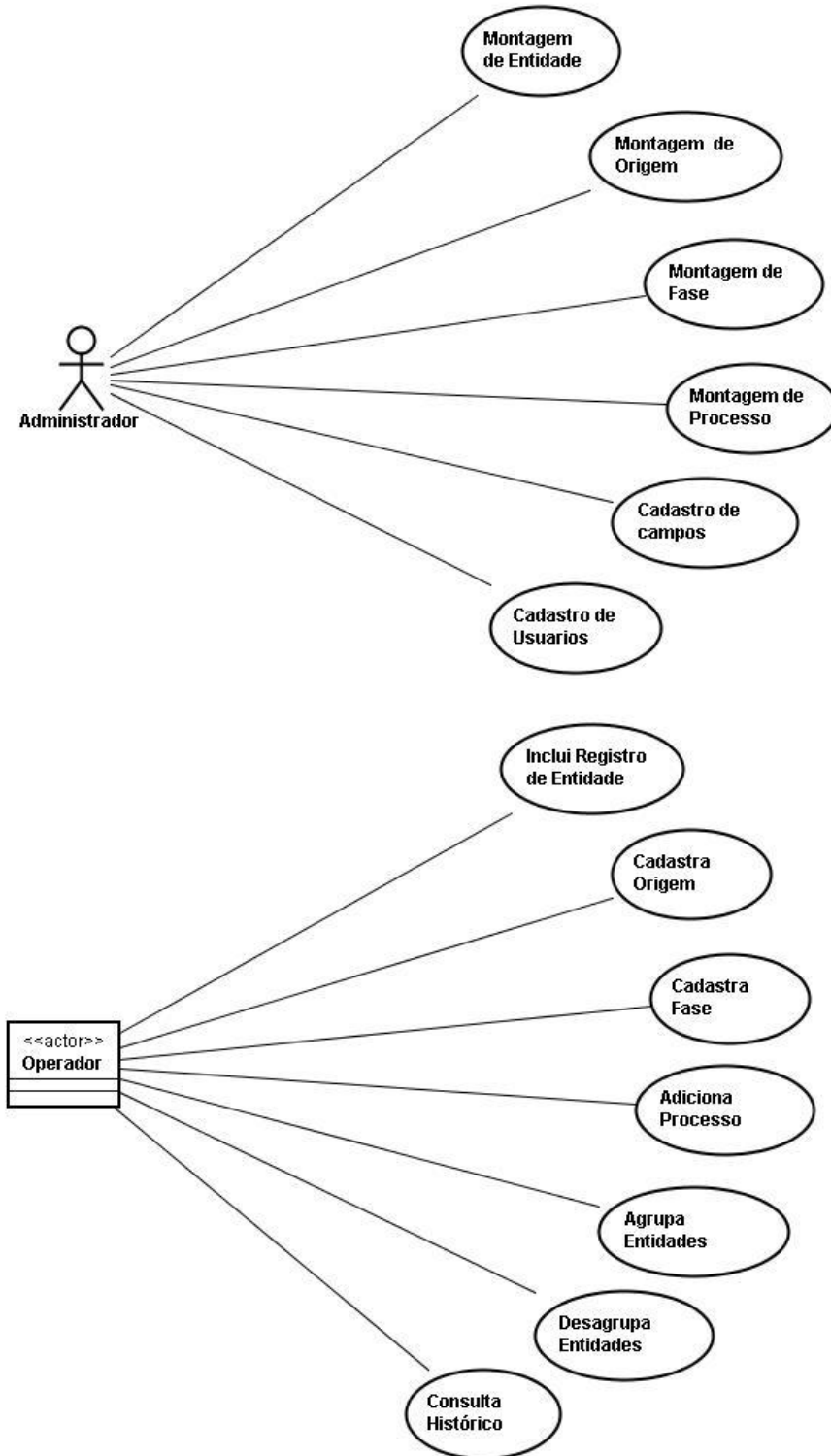
```

<ENTIDADES ID="Entidades">
  <ENTIDADE ID="Milho" CODIGO="6" ATIVO="N">
    <CAMPO CODIGOCAMPO="11" NOME="Peso" TPHTML="text" VALORES=";">
      15 kg
    </CAMPO>
    <CAMPO CODIGOCAMPO="46" NOME="Variedade" TPHTML="text" VALORES=";">
      milho branco
    </CAMPO>
    <ORIGEM ATIVO="S" nome="Origem" ID="Origem">
      <CAMPO CODIGOCAMPO="14" NOME="Propriedade" TPHTML="text" VALORES=";">
        Fazenda Canguiri
      </CAMPO>
      <CAMPO CODIGOCAMPO="15" NOME="Cooperado" TPHTML="text" VALORES=";">
        Ana Maria
      </CAMPO>
      <CAMPO CODIGOCAMPO="9" NOME="Matricula" TPHTML="text" VALORES=";">
        1236
      </CAMPO>
    </ORIGEM>
    <FASES ID="Fases">
      <FASE CODIGO="1" ATIVO="S" tipo="Fase" ID="Secagem">
        <CAMPO CODIGOCAMPO="26" NOME="Temperatuda medida" TPHTML="text" VALORES=";">
          30°
        </CAMPO>
        <PROCESSOS ID="Processos">
          <PROC_ADD CODIGO="" tipo="Processo" ATIVO="N" NOME="" />
        </PROCESSOS>
      </FASE>
      <FASE CODIGO="2" ATIVO="S" tipo="Fase" ID="Armazenamento">
        <CAMPO CODIGOCAMPO="8" NOME="Silo" TPHTML="text" VALORES=";">21</CAMPO>
        <CAMPO CODIGOCAMPO="42" NOME="Data de Entrada" TPHTML="text" VALORES=";">
          10/02/2006
        </CAMPO>
        <CAMPO CODIGOCAMPO="48" NOME="Data de saída" TPHTML="text" VALORES=";">
          20/02/2006
        </CAMPO>
        <PROCESSOS ID="Processos">
          <PROCESSO ID="Monitoramento" CODIGO="6">
            <CAMPO CODIGOCAMPO="25" NOME="Data" TPHTML="text" VALORES=";">
              15/02/2006
            </CAMPO>
            <CAMPO CODIGOCAMPO="26" NOME="Temperatuda medida" TPHTML="text" VALORES=";">
              26
            </CAMPO>
            <CAMPO CODIGOCAMPO="32" NOME="Código da Armadilha" TPHTML="text" VALORES=";">
              3
            </CAMPO>
            <CAMPO CODIGOCAMPO="36" NOME="Total de insetos" TPHTML="text" VALORES="0;">
              231
            </CAMPO>
            <CAMPO CODIGOCAMPO="35" NOME="Tribolium" TPHTML="text" VALORES="0;">
              56
            </CAMPO>
            <CAMPO CODIGOCAMPO="34" NOME="Rizoperta" TPHTML="text" VALORES="0;">
              32
            </CAMPO>
            <CAMPO CODIGOCAMPO="33" NOME="Citofilos" TPHTML="text" VALORES="0;">
              99
            </CAMPO>
          </PROCESSO>
        </PROCESSOS>
      </FASE>
    </ENTIDADE>
  </ENTIDADES>

```

APÊNDICE 2 - CASOS DE USO

Figura 03 – Diagrama de casos de uso



APÊNDICE 3 - DESCRIÇÃO RESUMIDA DOS CASOS DE USO

MONTAGEM DE ENTIDADE

Este caso visa a criação dos formulários de cadastro das Entidades do sistema. A responsabilidade de criação da tela fica a cargo do Administrador do sistema.

MONTAGEM DE TIPO DE ORIGEM

Este caso visa a criação dos formulários de cadastro de tipos de origem das entidades. A responsabilidade de criação da tela fica a cargo do Administrador do sistema.

MONTAGEM DE FASE

Este caso visa a criar os formulários de cadastro de fases das entidades. A responsabilidade de criação da tela fica a cargo do Administrador do sistema.

MONTAGEM DE PROCESSO

Este caso visa criar os formulários de cadastro de processos das fases. A responsabilidade de criação da tela fica a cargo do Administrador do sistema

CADASTRO DE CAMPOS

Este caso de uso visa a criação dos campos que serão utilizados pelo administrador para compor seus formulários. A responsabilidade de criação e uso dos campos fica a cargo do Administrador do sistema.

CADASTRO DE USUÁRIOS

Este caso de uso visa cadastrar os usuários que irão utilizar o sistema. A responsabilidade pelo cadastramento de usuários fica a cargo do Administrador.

INCLUI REGISTRO DE ENTIDADE

Este caso tem por objetivo alimentar com dados os formulários de Entidades criados pelo administrador, ou seja, cadastrar as entidades que farão parte do sistema e serão rastreadas posteriormente. A responsabilidade cadastrar entidades fica a cargo do Operador do sistema.

CADASTRA ORIGEM

Este caso tem por objetivo alimentar com dados o formulário de Origem criado pelo administrador, ou seja, cadastrar a origem de cada uma das entidades que farão parte do sistema. A responsabilidade cadastrar origens fica a cargo do Operador do sistema.

CADASTRA FASE

Este caso tem por objetivo alimentar com dados os formulários das fases, criados pelo administrador. A responsabilidade cadastrar fases fica a cargo do Operador do sistema.

ADICIONA PROCESSO

Este caso tem por objetivo alimentar com dados os formulários de processos criados pelo administrador, ou seja, adicionar em cada fase os processos ocorridos com as entidades que farão parte do sistema. A responsabilidade cadastrar processos fica a cargo do Operador do sistema.

AGRUPA ENTIDADES

Este caso tem por objetivo agrupar entidades do mesmo tipo que, em algum momento da produção, venham a se juntar fisicamente. A partir deste agrupamento passarão a ser tratadas com uma única identificação construindo um histórico comum. A responsabilidade de agrupar entidades fica a cargo do Operador do sistema.

DESAGRUPA ENTIDADES

Este caso tem por objetivo desagrupar uma entidade que, em algum momento da produção, venha a ser dividida em grupos menores. A partir deste desagrupamento passarão a ser tratados como entidades distintas, construindo um histórico individual. A responsabilidade de desagrupar entidades fica a cargo do Operador do sistema.

CONSULTA HISTÓRICO

Este caso tem o objetivo de consultar os dados sobre cada uma das entidades cadastradas no sistema, realizando assim a Rastreabilidade destas entidades. A responsabilidade consultar histórico fica a cargo do Operador do sistema.

APÊNDICE 4 - FLUXO DE EVENTOS DOS CASOS DE USO

MONTAGEM DE ENTIDADE

Pré-requisitos: Os campos já devem ter sido criados

Fluxo principal

O Administrador entra na tela Cadastro de Entidade. Ele escreve o nome da entidade. Adiciona os campos que deverão fazer parte da tela, seleciona um tipo de origem e a lista de fases que farão parte do histórico desta entidade. Clica no botão Salvar para gravar as informações da tela de Entidade no arquivo de telas.

Fluxos alternativos

Alterar tela

Abre a Lista de Entidade e clica na opção editar da tela desejada, o sistema abre o formulário de edição da entidade com os dados já cadastrados, o administrador faz as alterações e clica em Salvar.

Excluir tela

Abre a Lista de Entidade e clica na opção Apagar da tela desejada, o sistema não irá remover o registro da tela do arquivo de entidades, apenas tornará o registro inativo.

MONTAGEM DE ORIGEM

Pré-requisitos: Os campos já devem ter sido criados

Fluxo principal

O Administrador entra na tela Cadastro de tipo Origem. Escreve o nome da origem e adiciona os campos que deverão fazer parte da tela, clica no botão Salvar para gravar as informações da tela de Origem no arquivo de telas.

Fluxos Alternativos

Alterar tela

Abre a Lista de Origens e clica na opção editar da tela desejada, o sistema abre o formulário de Edição da origem, o administrador faz as alterações e clica em Salvar.

Excluir tela

Abre a Lista de Origens e clica na opção Apagar da tela desejada, o sistema não irá remover o registro da tela do arquivo de origem, apenas tornará o registro inativo.

CRIA TELA FASE

Pré-requisitos: Os campos já devem ter sido criados

Fluxo principal

O administrador entra na tela Cadastro de Fase. Ele escreve o nome da fase, adiciona os campos que deverão fazer parte da tela e a lista de processos que poderão ocorrer nesta fase. Clica no botão Salvar para gravar as informações da tela de fases no arquivo de telas.

Fluxos Alternativos

Alterar tela

Abre a Lista de fases e clica na opção editar da tela desejada, o sistema abre o formulário de edição da fase com os dados já cadastrados, o administrador faz as alterações e clica em Salvar.

Excluir tela

Abre a Lista de fases e clica na opção Apagar da tela desejada, o sistema não irá remover o registro da tela do arquivo de fases, apenas tornará o registro inativo.

CRIA TELA PROCESSO

Pré-requisitos: Os campos já devem ter sido criados

Fluxo principal

O administrador entra na tela Cadastro de Processos. Ele escreve o nome do processo, adiciona os campos que deverão fazer parte da tela. Clica no botão Salvar para gravar as informações da tela de processos no arquivo de telas.

Fluxos Alternativos

Alterar tela

Abre a Lista de Processos e clica na opção editar da tela desejada, o sistema abre o formulário de edição do processo com os dados já cadastrados, o administrador faz as alterações e clica em Salvar.

Excluir tela

Abre a Lista de Processos e clica na opção Apagar da tela desejada, o sistema não irá remover o registro da tela do arquivo de processos, apenas tornará o registro inativo.

CRIA CAMPOS

Pré-requisitos: Nenhum

Fluxo principal

O Administrador preenche os campos do formulário Cadastro de Campos e clica no botão Salvar para gravar o campo no arquivo de campos. Na mesma tela do Cadastro de Campos aparecerá a Lista de Campos, com os campos já cadastrados.

Fluxos Alternativos

Alterar:

Na a Lista de campos o administrador deverá clicar sobre o campo desejado, o sistema irá carregar os dados do campo no formulário de cadastro. O administrador irá fazer as alterações que desejar e clicar no botão Salvar.

Excluir:

Na lista de campos o administrador deverá clicar no link Apagar, correspondente ao campo que deverá ser apagado. O registro não será excluído do arquivo de campos, apenas será inativado, para que as telas que já possuem este campo continuem a utilizá-lo.

CADASTRO DE USUÁRIOS

Pré-requisitos: Nenhum

Fluxo principal

O Administrador abre a tela de Cadastro de usuário, preenche os campos do formulário clica no botão Salvar. Na mesma tela do Cadastro de Usuários aparecerá a Lista de Usuários, com os usuários já cadastrados. Estes usuários serão as pessoas que terão acesso ao sistema.

Fluxos Alternativos

Alterar:

Na lista de usuários o administrador deverá escolher um usuário e abrir o seu cadastro, fazer as alterações que desejar e clicar no botão Alterar.

Excluir:

Na lista de usuários o administrador deverá escolher um usuários e clicar no link Apagar e confirmar a exclusão.

CADASTRAR ENTIDADE

Pré-requisitos: A tela de Entidade já deve ter sido criada.

Fluxo principal

O operador escolhe tipo de entidade que deseja trabalhar. Clica na opção Novo Registro, preenche as informações da entidade e clica em Salvar para gravar os dados no arquivo.

Fluxos Alternativos

Alterar

Para alterar o cadastro de uma entidade o operador deverá selecionar a entidade que deseja alterar no menu lateral. Clicar no link Editar, fazer as alterações e clicar em Alterar.

Excluir

Para excluir uma Entidade o operador deverá entrar no seu cadastro e clicar em Excluir.

Exibir Relatório completo

O operado escolhe a entidade desejada e clica no link Relatório completo, o sistema apresentará o relatório com todos os dados referentes àquela entidade.

CADASTRAR ORIGEM

Pré-requisitos: Que já exista uma Entidade cadastrada e a tela de origem já tenha sido criada.

Fluxo principal

Primeiro deverá selecionar a Entidade na qual será adicionada a origem no menu lateral, clicar no link Origem sob o nome da entidade. Clicar no link Cadastrar, preencher os dados da origem e clicar em Salvar. Assim os dados da origem serão incluídos no registro da entidade a que ela pertence.

Fluxos Alternativos

Alterar

Escolhe a entidade no menu lateral e depois clica na Origem. Clica no link Editar, o sistema abre o modo de edição, o operador altera os dados e clica em Alterar.

CADASTRAR FASE

Pré-requisitos: Que já exista uma tela de fase e já exista um entidade cadastrada.

Fluxo principal

O operador escolhe na lista uma Entidade à qual deseja adicionar a fase. Escolhe, na entidade, a fase na qual deseja cadastrar os dados. Clica no link Cadastrar, preenche os dados da fase e clica na opção Salvar. Assim os dados da fase serão incluídos no arquivo da entidade a que ela pertence.

Fluxos Alternativos

Alterar

Escolhe primeiro a entidade e depois clica na fase que deseja alterar. Clica no link Editar, o sistema abre o modo de edição, o operador altera os dados e clica em Alterar.

CADASTRAR PROCESSO

Pré-requisitos: Já deve existir ao menos uma tela de processo e já ter sido cadastrada uma entidade.

Fluxo principal

O operador seleciona a entidade e em seguida a fase em que deseja adicionar o processo. No menu lateral, sob a fase, clica no link Processos, seleciona o processo que deseja cadastrar. Preenche os dados do processo e clica em Salvar.

Fluxos Alternativos

Alterar

Para alterar o processo o operador deverá escolher a Entidade e em seguida a Fase do processo no menu lateral. Clicar sobre o processo desejado, clicar no link Editar, fazer as alterações e clicar em Alterar.

Excluir

Para alterar o processo o operador deverá escolher a Entidade e em seguida a Fase do processo no menu lateral. Clicar sobre o processo desejado e clicar no link Excluir.

AGRUPAR ENTIDADES

Pré-requisitos: Que existam entidades já cadastradas.

Fluxo principal

O operador deverá criar um registro da nova entidade, originada do agrupamento de outras entidades. Após criar o registro da nova entidade o operador clica no link Agrupar. O sistema irá trazer um menu com todas as entidades que podem ser agrupadas. O operador irá selecionar as entidades que compõe este agrupamento e clicar em agrupar.

A partir daí a origem desta nova entidade será chamada de 'Agrupamento', e conterà links para os registros das entidades que a geraram. As entidades agrupadas passarão para um status de Inativas, não podendo mais sofrer exclusão, além disso receberão uma nova fase, chamada 'Agrupamento', com um link para a nova entidade.

DESAGRUPAR ENTIDADES

Pré-requisitos: Que existam entidades já cadastradas.

Fluxo principal

O operador escolhe a entidade que deseja desagrupar, no menu lateral. Clica no link Desagrupar, preenche os dados referentes ao desagrupamento e clica no botão Desagrupar.

O sistema colocará a entidade em um estado de Inativa. Incluirá em seu registro uma fase chamada 'Agrupamento', com links para as novas entidades geradas, e não mais permitirá sua exclusão. Serão então geradas novas entidades, originadas do agrupamento, cuja Origem será chamada de 'Agrupamento' e conterà um link para a entidade original.

CONSULTA HISTÓRICO

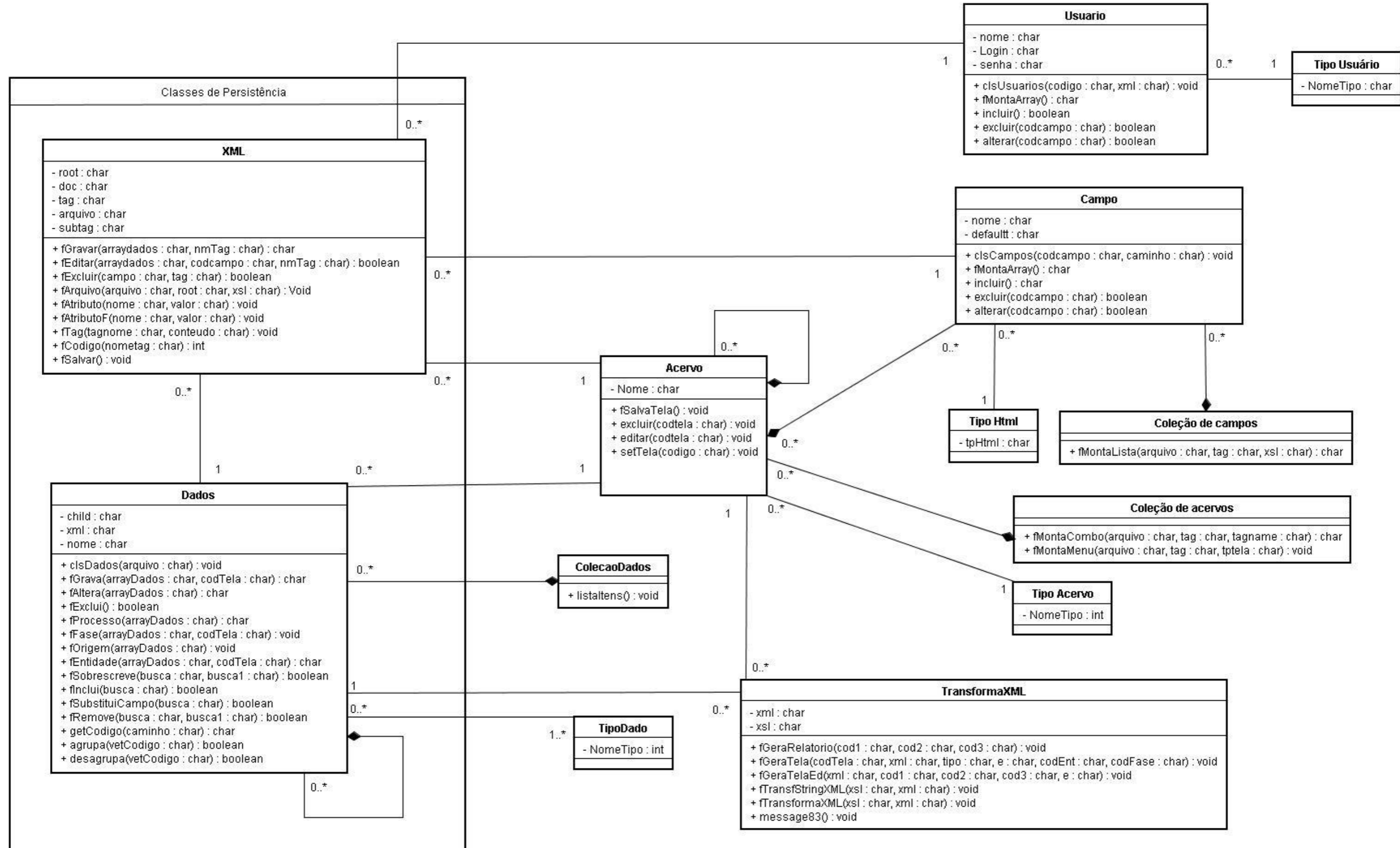
Pré-requisitos: Que existam entidades cadastradas.

Fluxo principal

O operador abre a Lista de Entidades e clica sobre uma entidade cadastrada. O sistema mostra a tabela com o histórico do item.

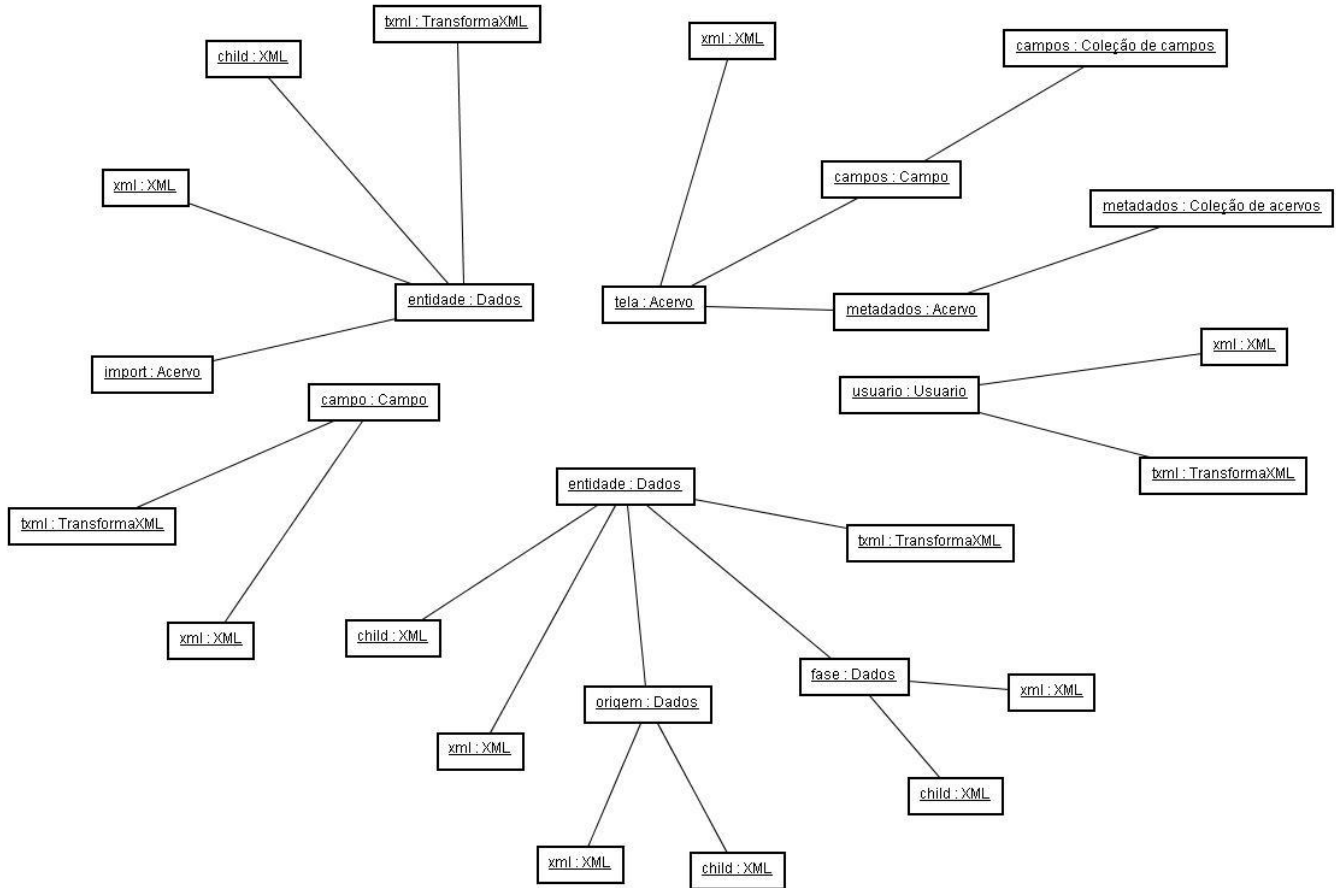
APÊNDICE 5 - DIAGRAMA DE CLASSES

Figura 04 – Diagrama de classes



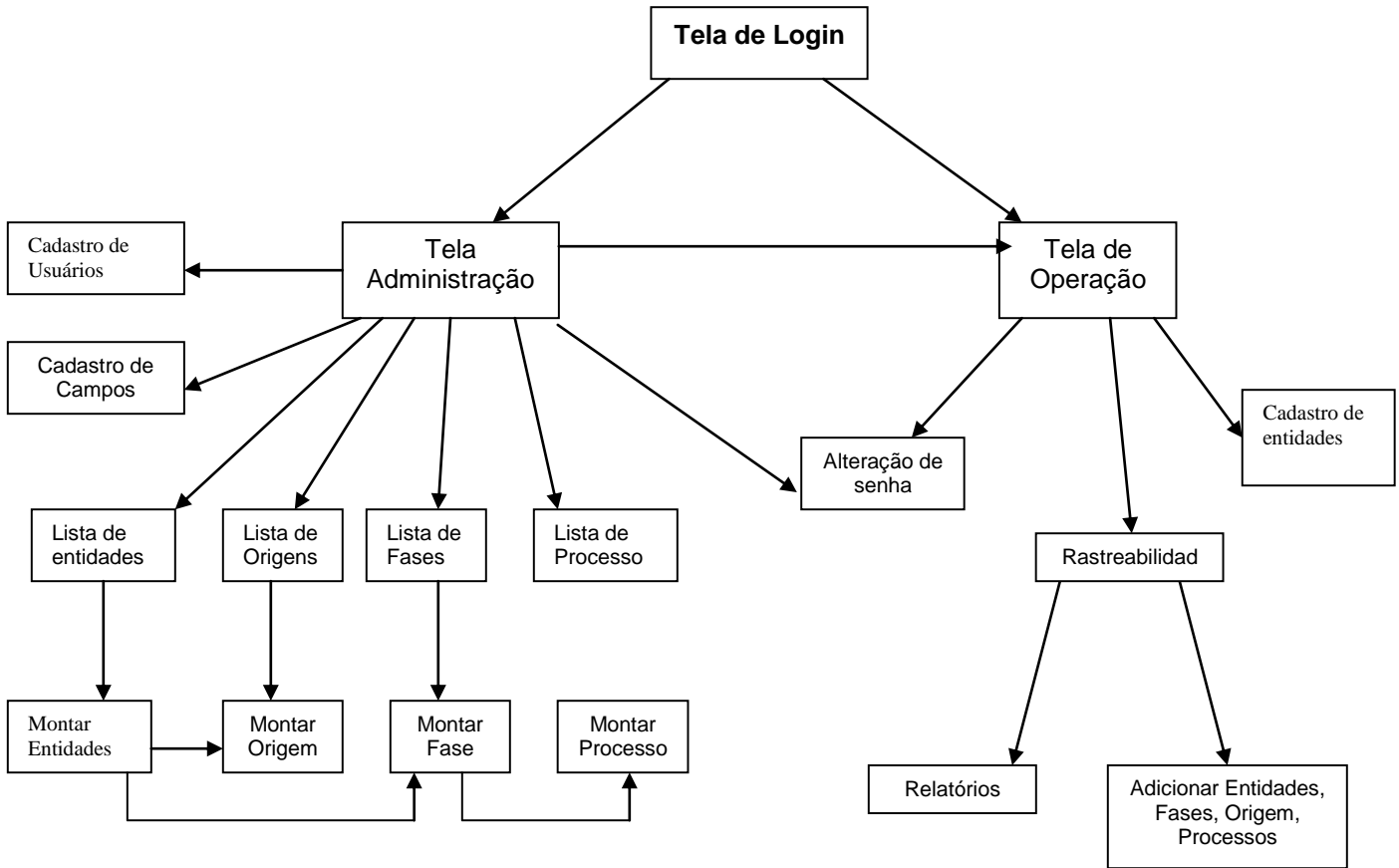
APÊNDICE 6 - DIAGRAMAS DE OBJETOS

Figura 05 – Diagrama de Objetos



APÊNDICE 6 - DIAGRAMA DE TELAS

Figura 06 – Diagrama de telas



APÊNDICE 7 – DIAGRAMAS DE SEQUÊNCIA E COLABORAÇÃO

Figura 07 – Tela cadastro de usuários

Cadastro de usuários Principal

Nome:

Tipo de Usuário:

Login:

Senha:

Lista de usuários				
	Nome	Login	Tipo de usuário	
1.	ANA MARIA PEREIRA	ana	Administrador	Apagar
2.	Ana	12	Usuário	Apagar
3.	EDUARDO RAFAEL	duda	Administrador	Apagar

Figura 08 – Seqüência Incluir Usuário

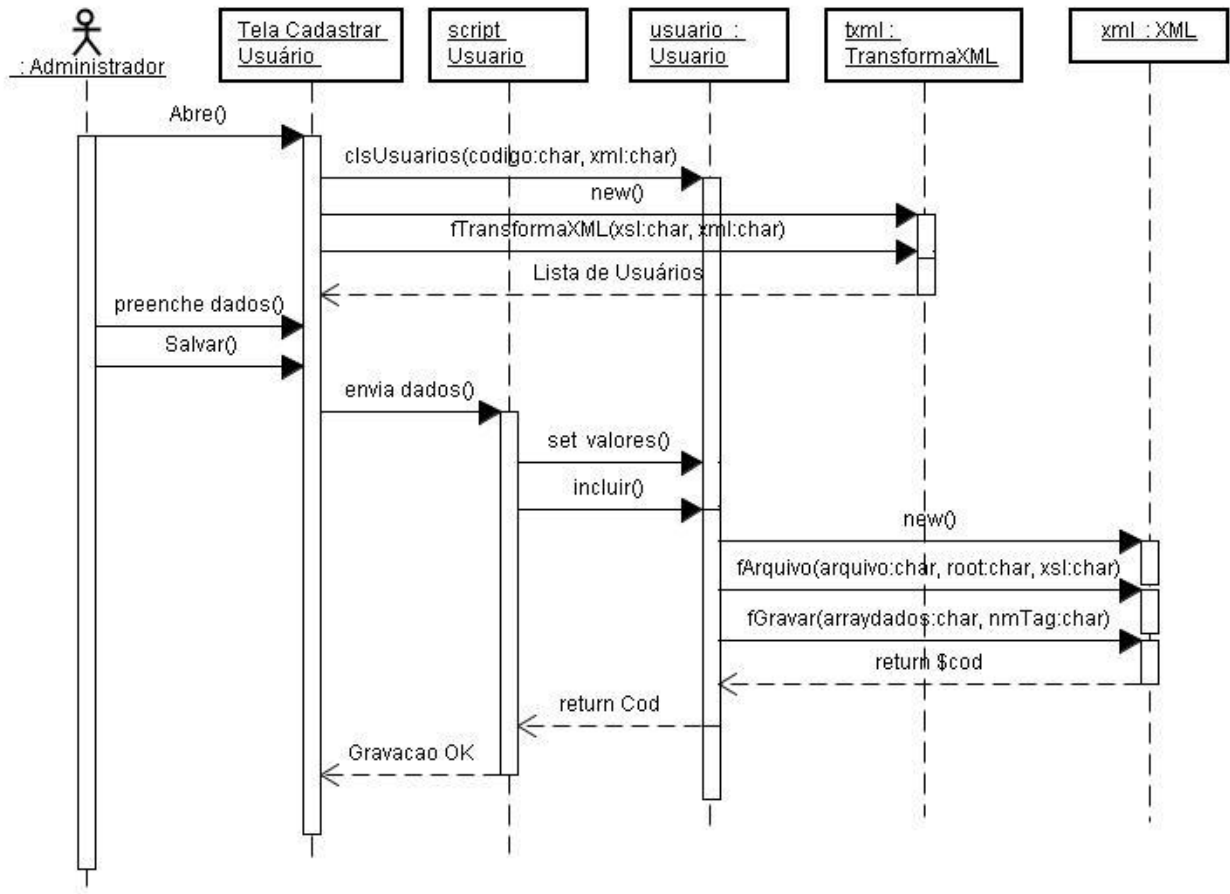


Figura 09 – Seqüência alterar Usuário

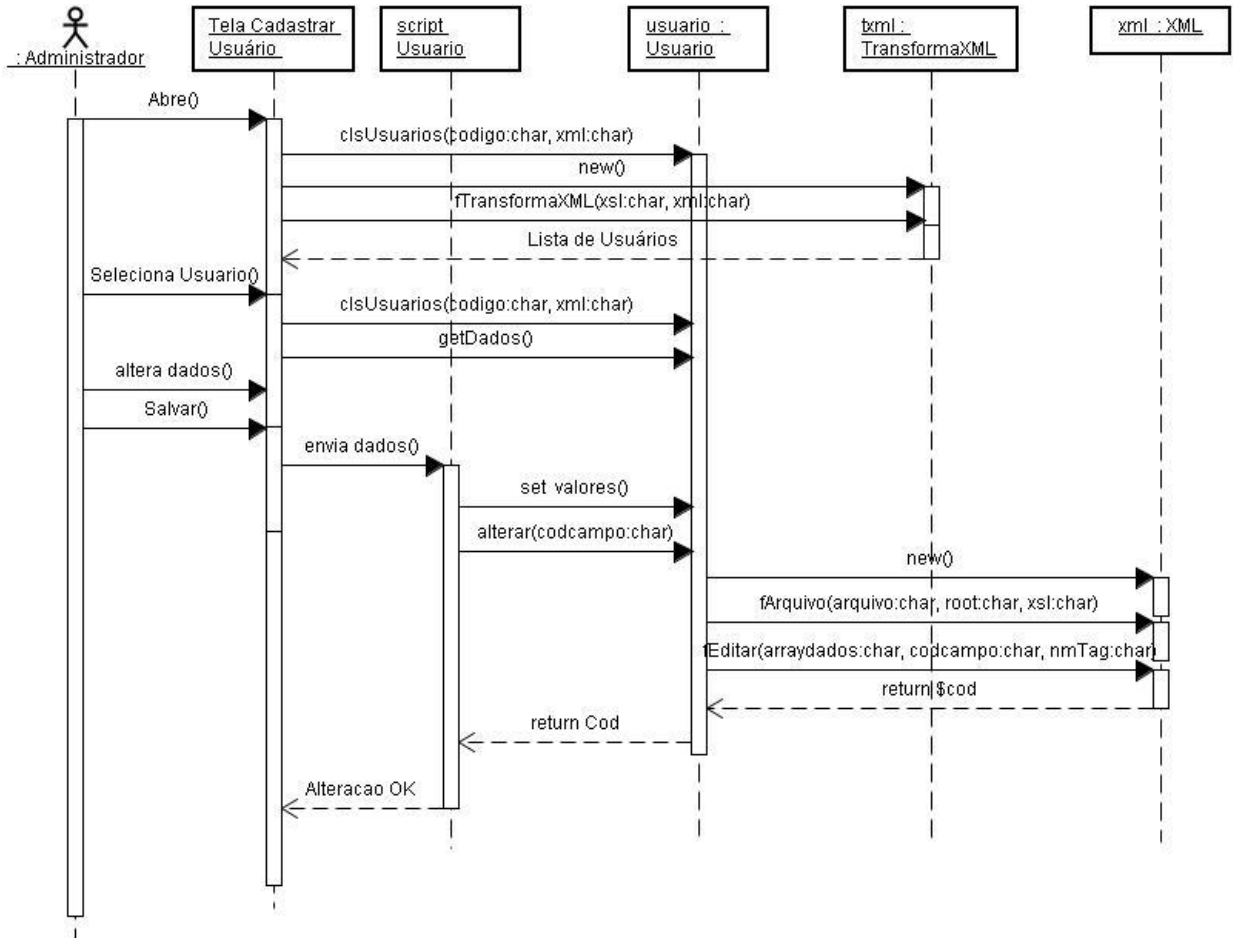


Figura 10 – Seqüência excluir Usuário

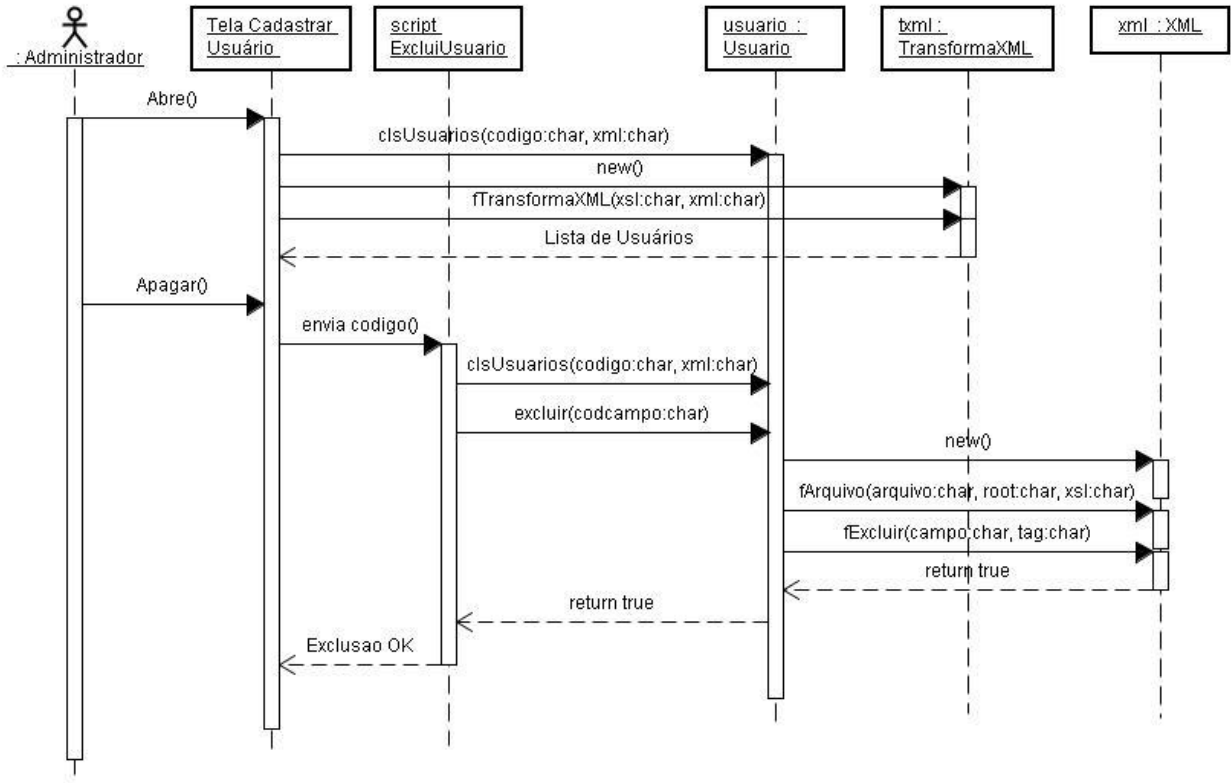


Figura 11 – Colaboração Incluir Usuário

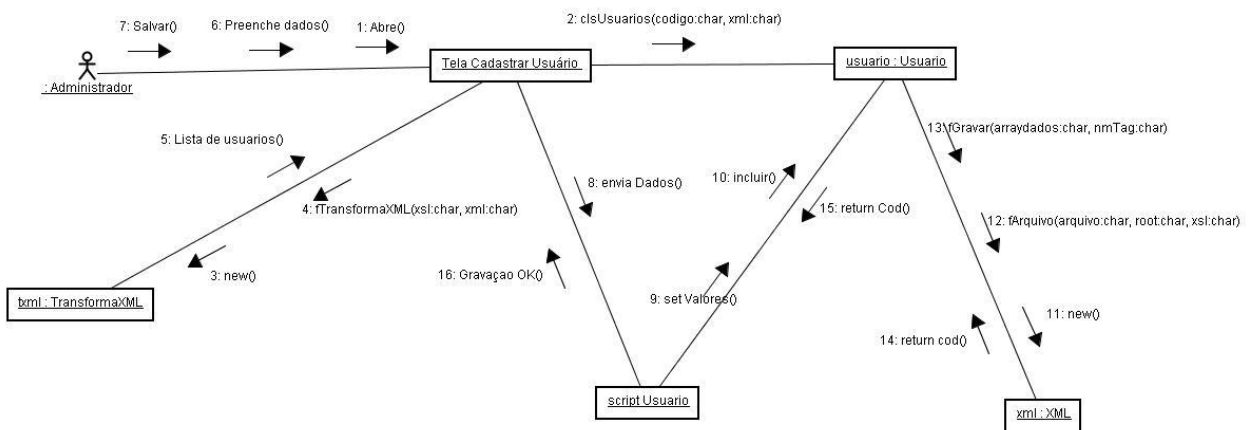


Figura 12 – Colaboração alterar Usuário

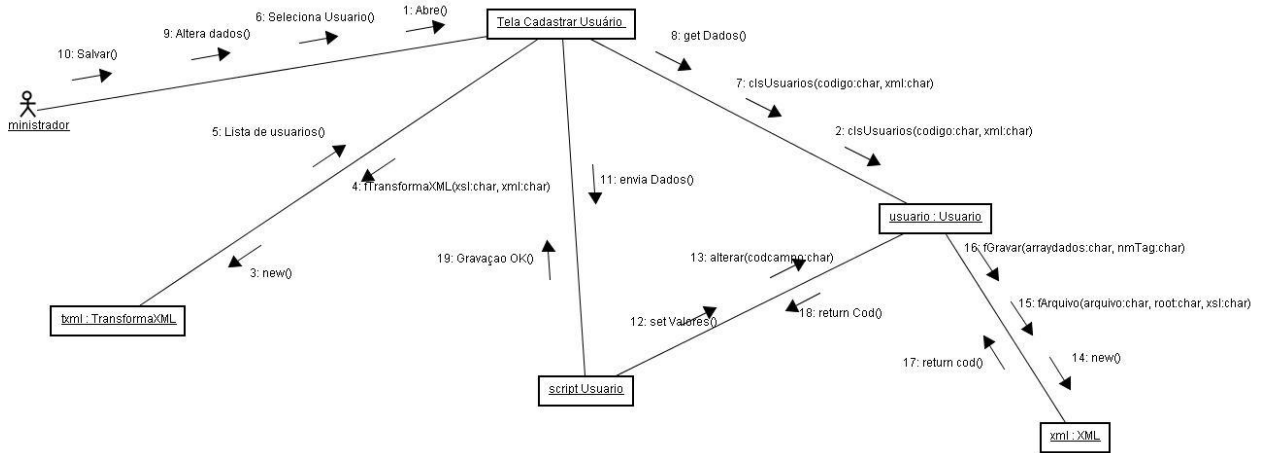


Figura 13 – Colaboração excluir Usuário

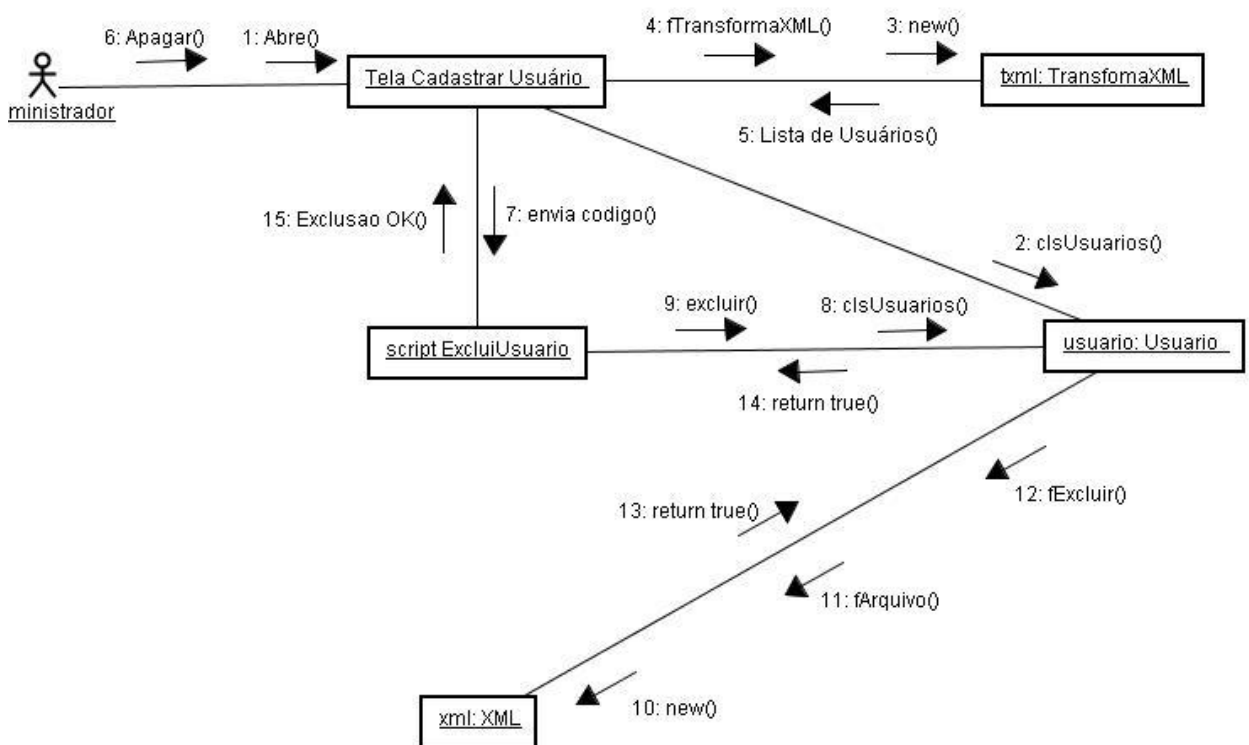


Figura 14 – Tela cadastro de campos

Cadastro de Campos Principal

Nome:

Tipo HTML:

Valor Default:

Lista de campos				
	Nome	Tipo de html	Valor Default	
1.	CEP	text		Apagar
2.	Nome	text	Eduardo	Apagar
3.	Observações	textarea		Apagar
4.	Ocorrências	text	Não houveram ocorrências	Apagar
5.	Período	select	Diurno Noturno	Apagar
6.	Produtos Químicos	select	Herbicida Fungicida Fertilizante	Apagar
7.	Responsável	text		Apagar
8.	Semente	checkbox	Milho tipo 1 Milho tipo 2	Apagar
9.	Telefone	text	Sem telefone	Apagar
10.	testes	text		Apagar

Webmasters: Ana Maria Pereira e Eduardo Rafael Coutinho dos Santos

Figura 15 – Seqüência Incluir Campo

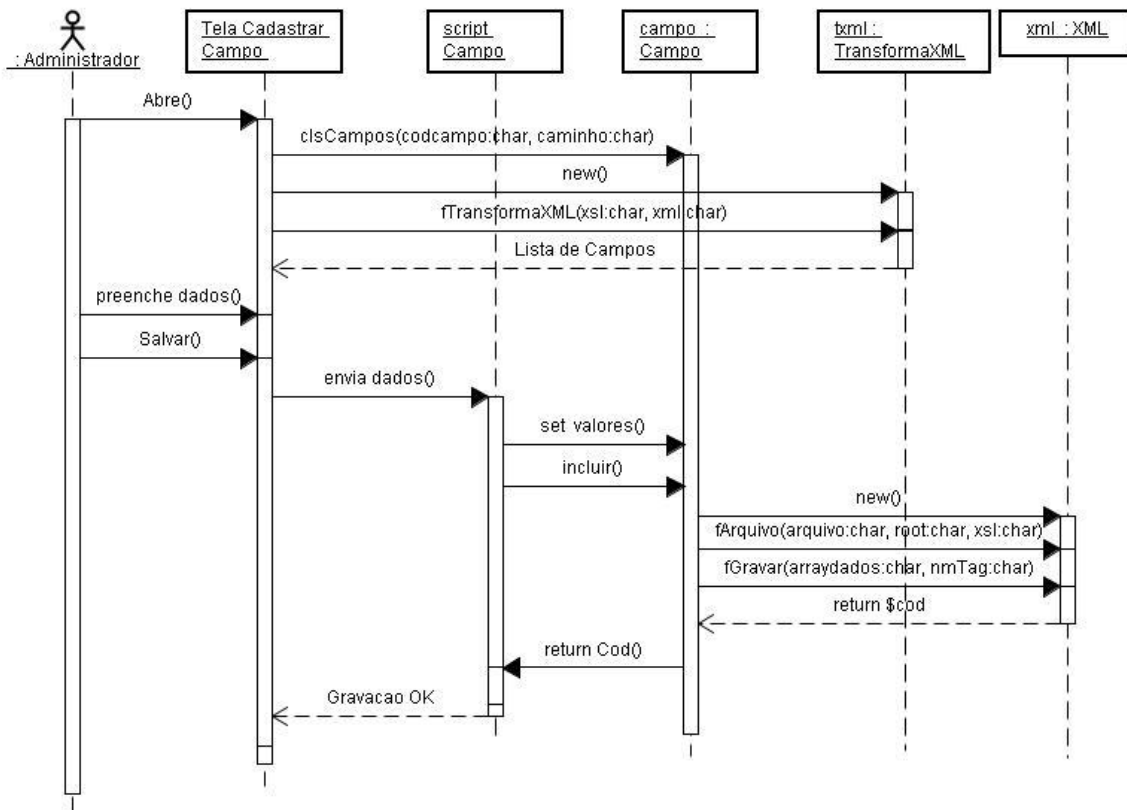


Figura 16 – Seqüência Alterar Campo

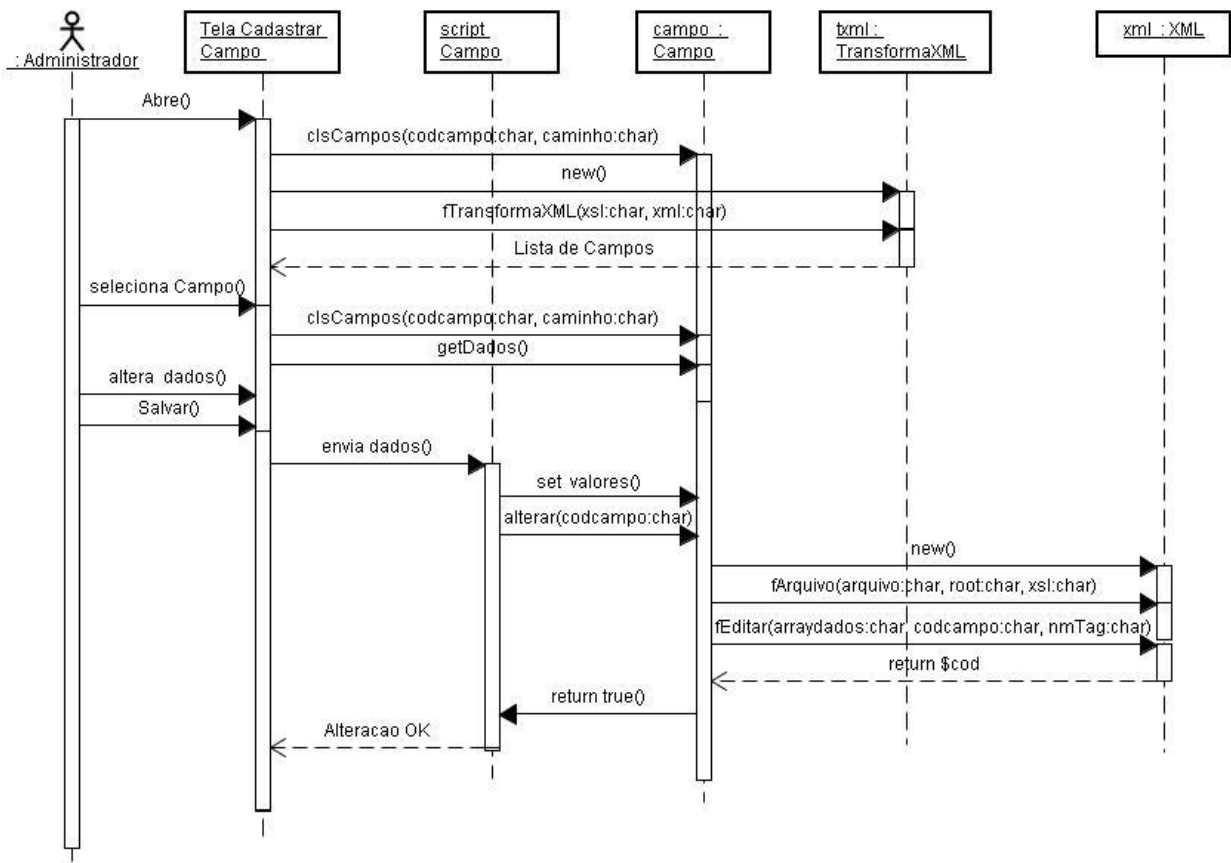


Figura 17 – Seqüência Excluir Campo

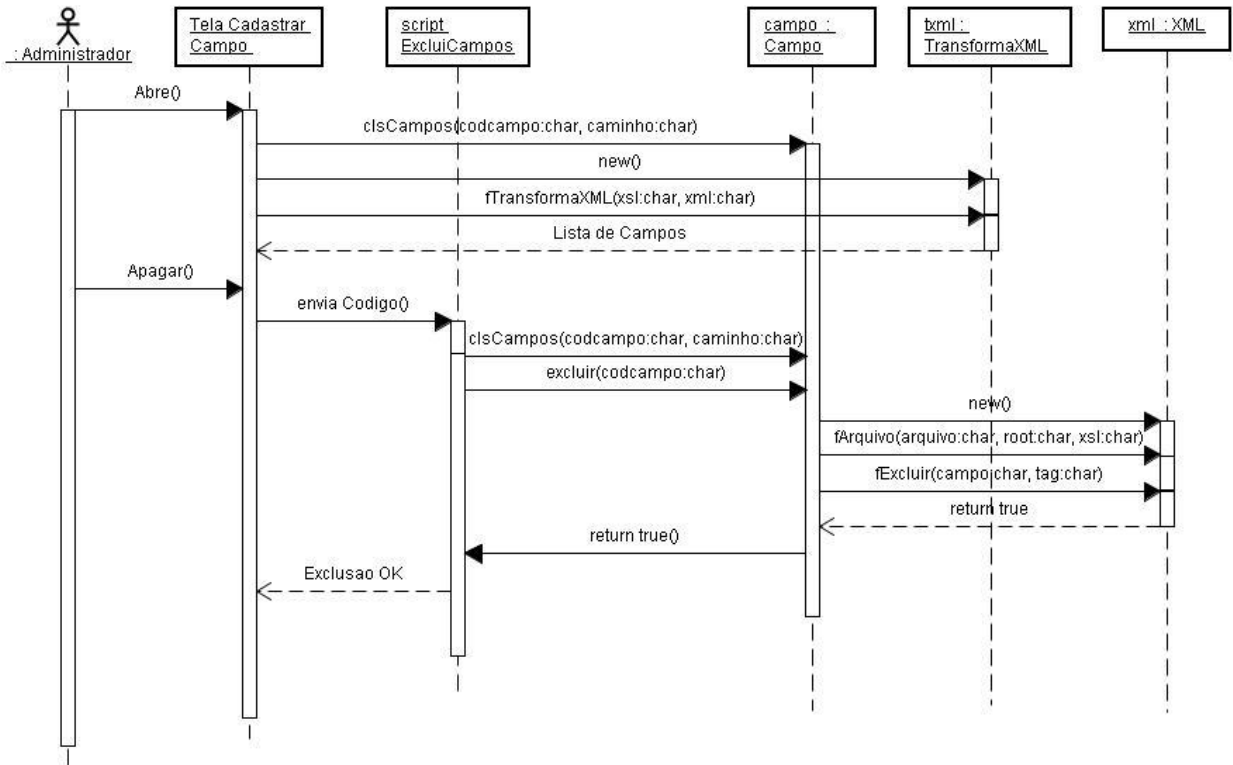


Figura 18 – Colaboração Incluir Campo

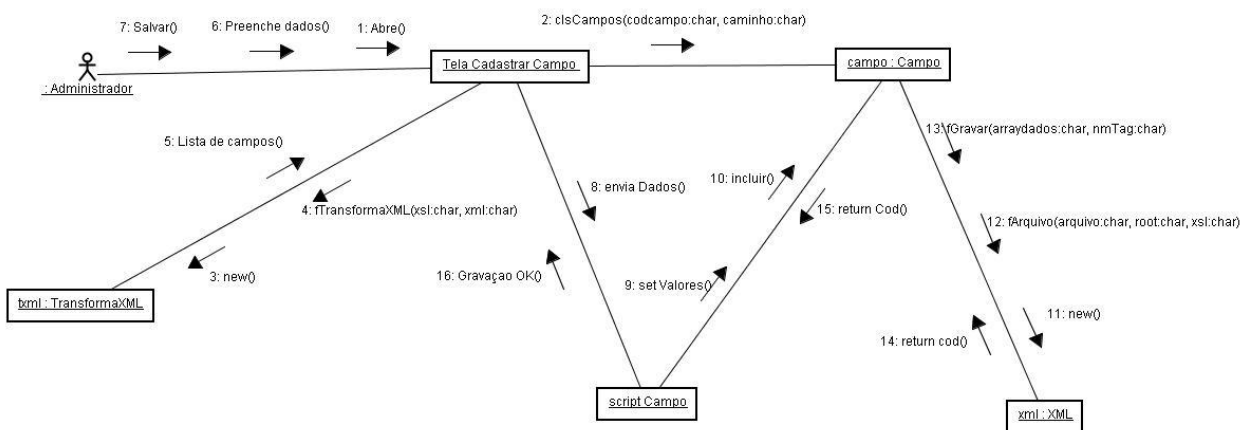


Figura 19 – Colaboração Alterar Campo

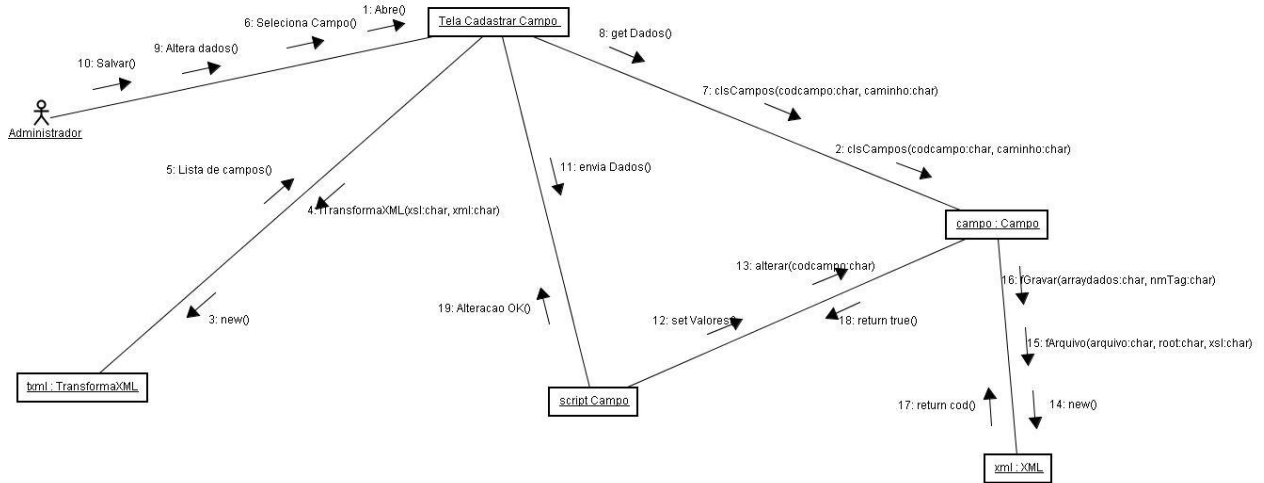


Figura 20 – Colaboração Excluir Campo

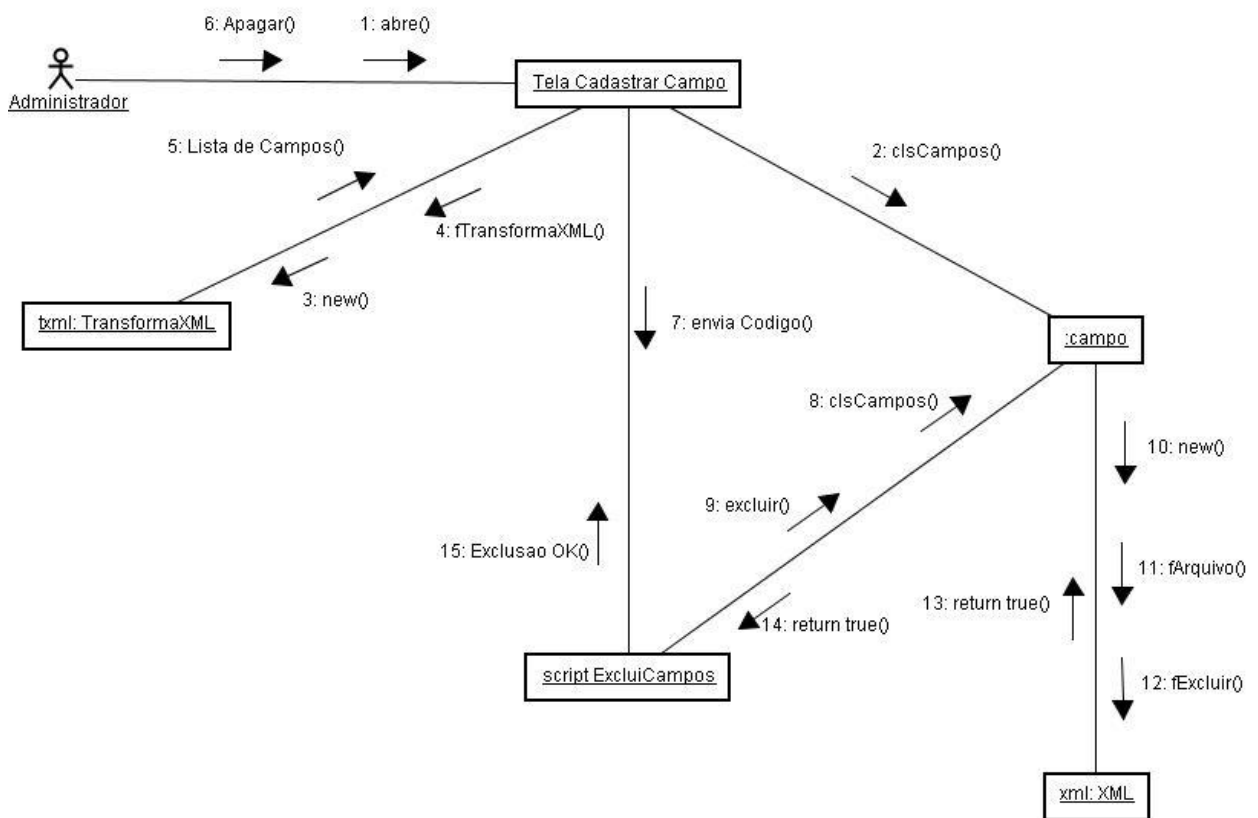


Figura 21 - Lista de Entidades

Lista de Entidades		
Nova		
Milho	Editar	Apagar
Ovelha	Editar	Apagar

Figura 22 – Tela Montar entidade

MONTAGEM DE ENTIDADE Voltar

NOME DA ENTIDADE: Salvar

Campo: Adicionar

Campos

Período	<input type="text" value="Diurno"/>	Excluir
Nome	<input type="text" value="Eduardo"/>	Excluir

Tipo de Origem

Lista de Fases Adicionar

Transporte [Excluir](#)

Figura 23 – Seqüência Cadastrar Entidades

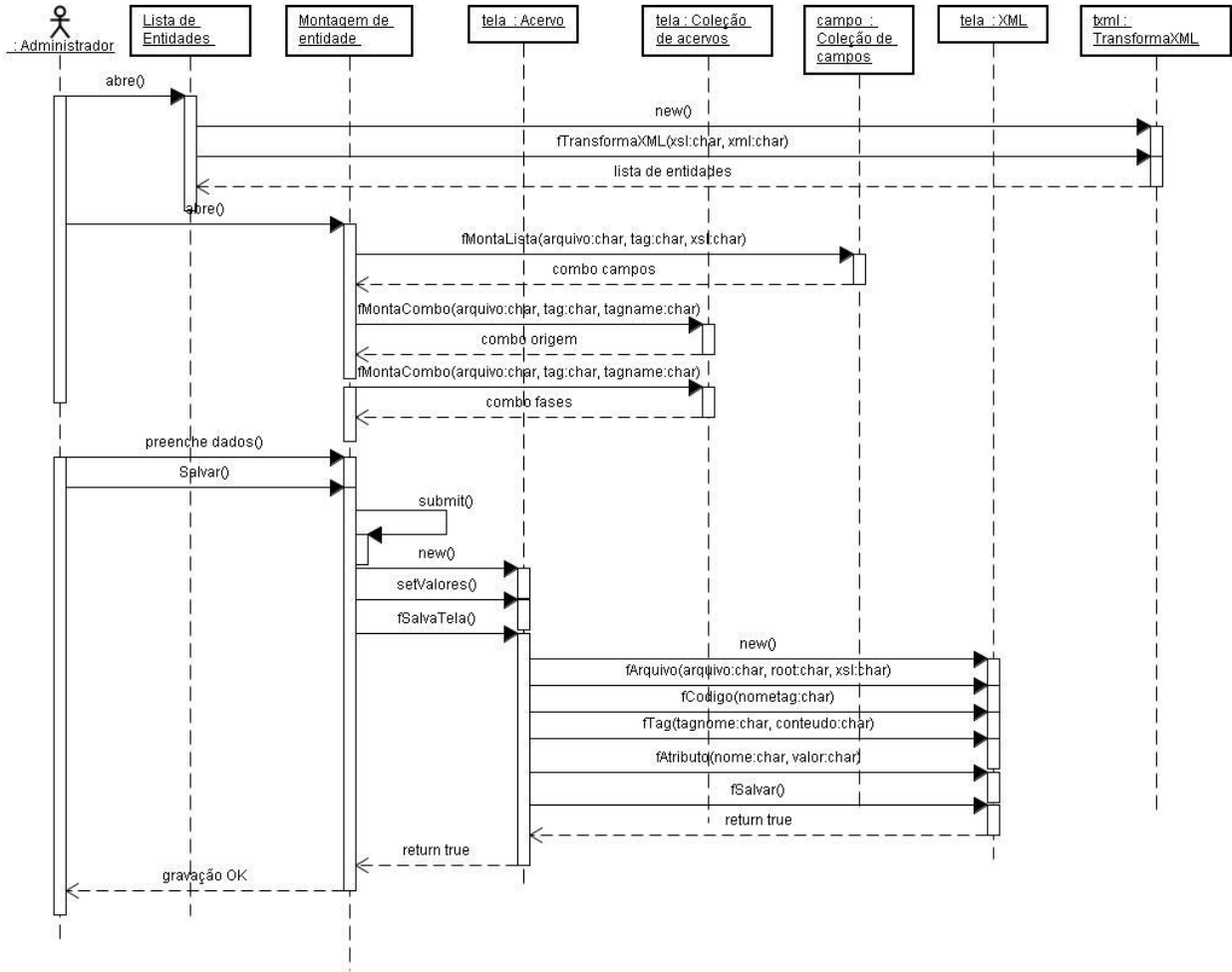


Figura 24 – Seqüência Alterar Entidade

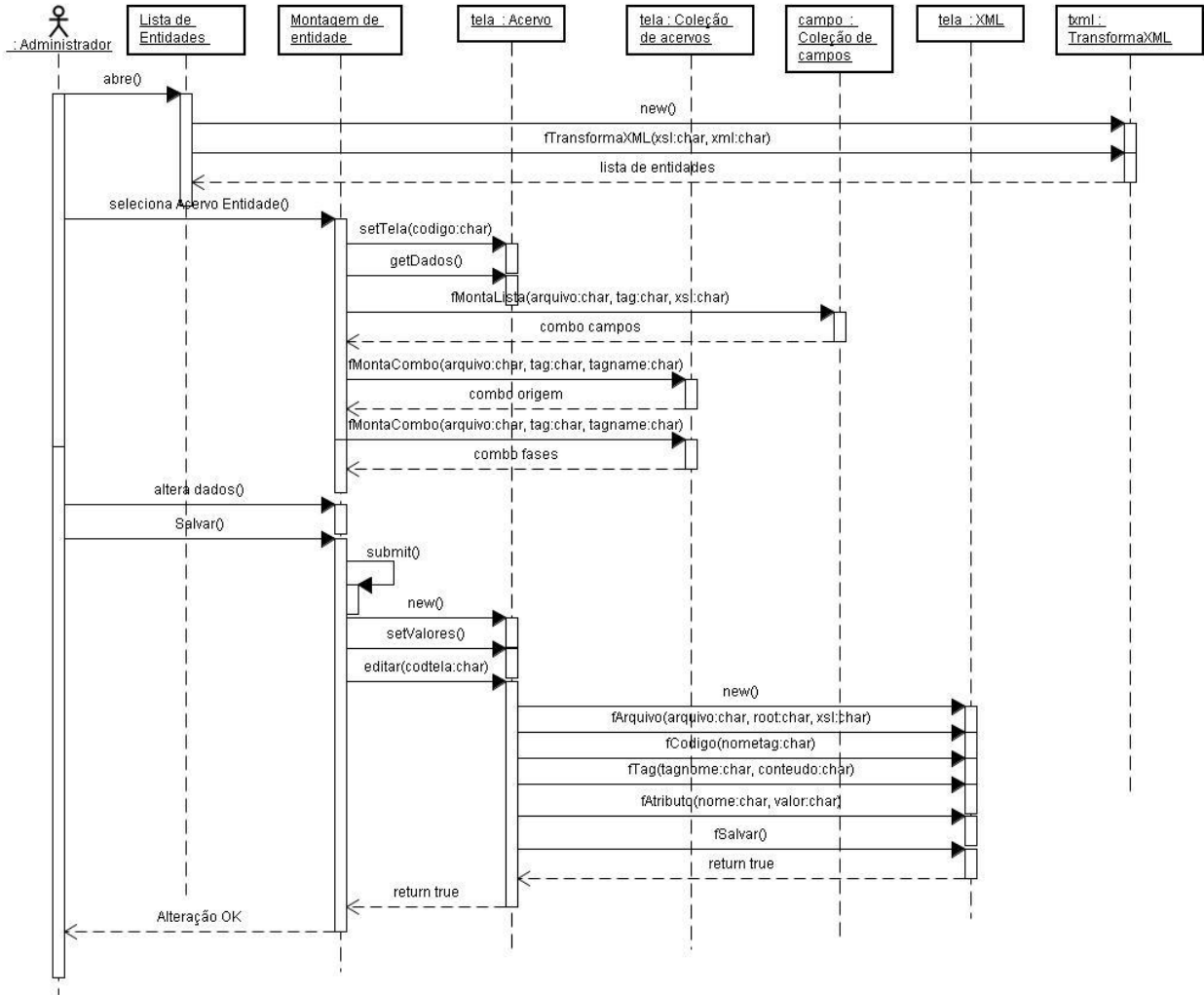


Figura 25 – Seqüência excluir Entidade

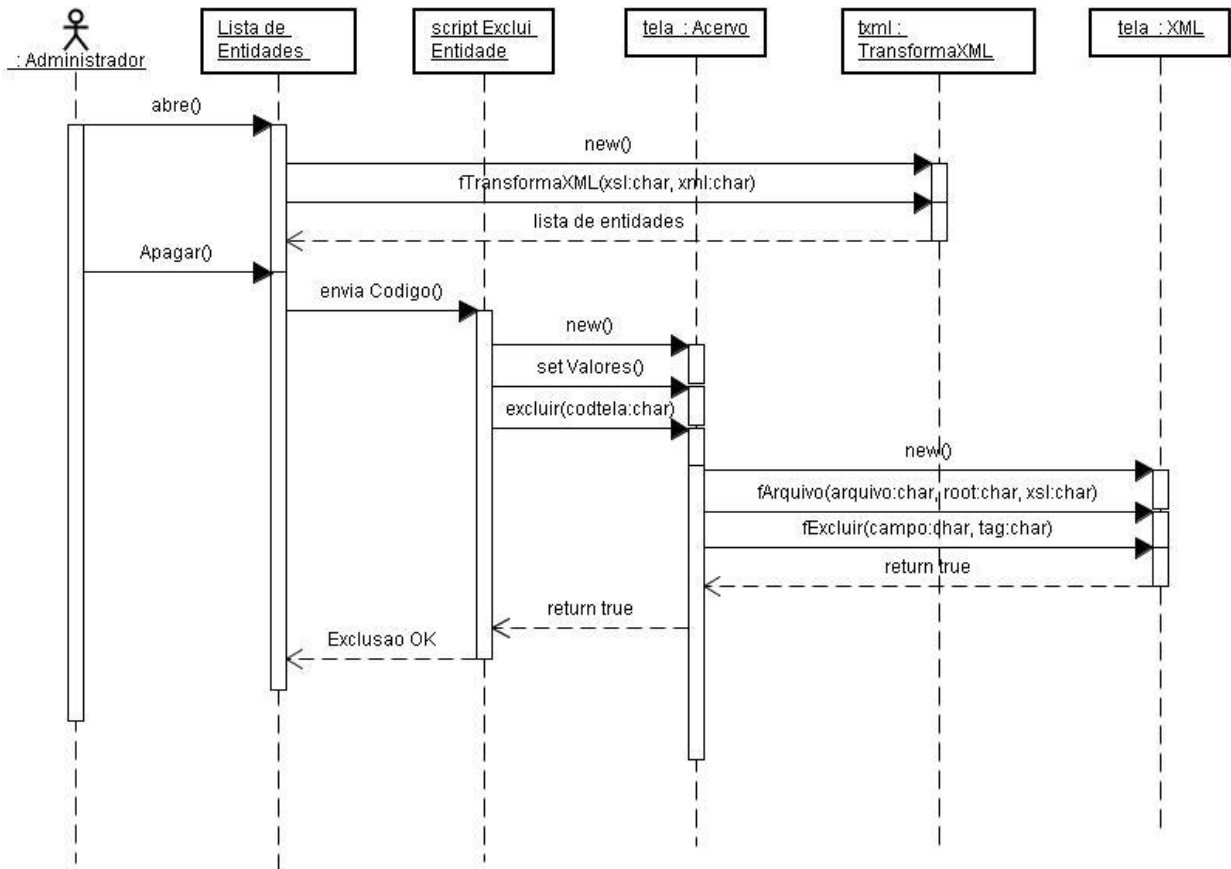


Figura 26 – Colaboração Cadastrar Entidades

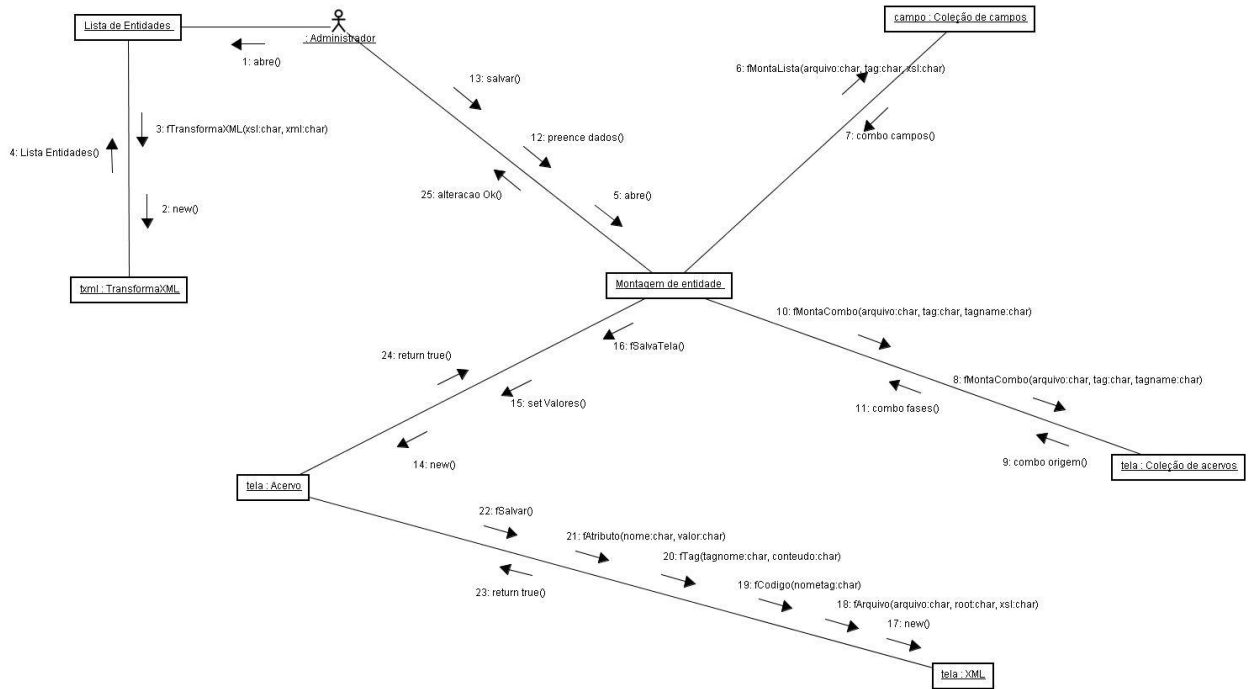


Figura 27 – Colaboração Alterar Entidade

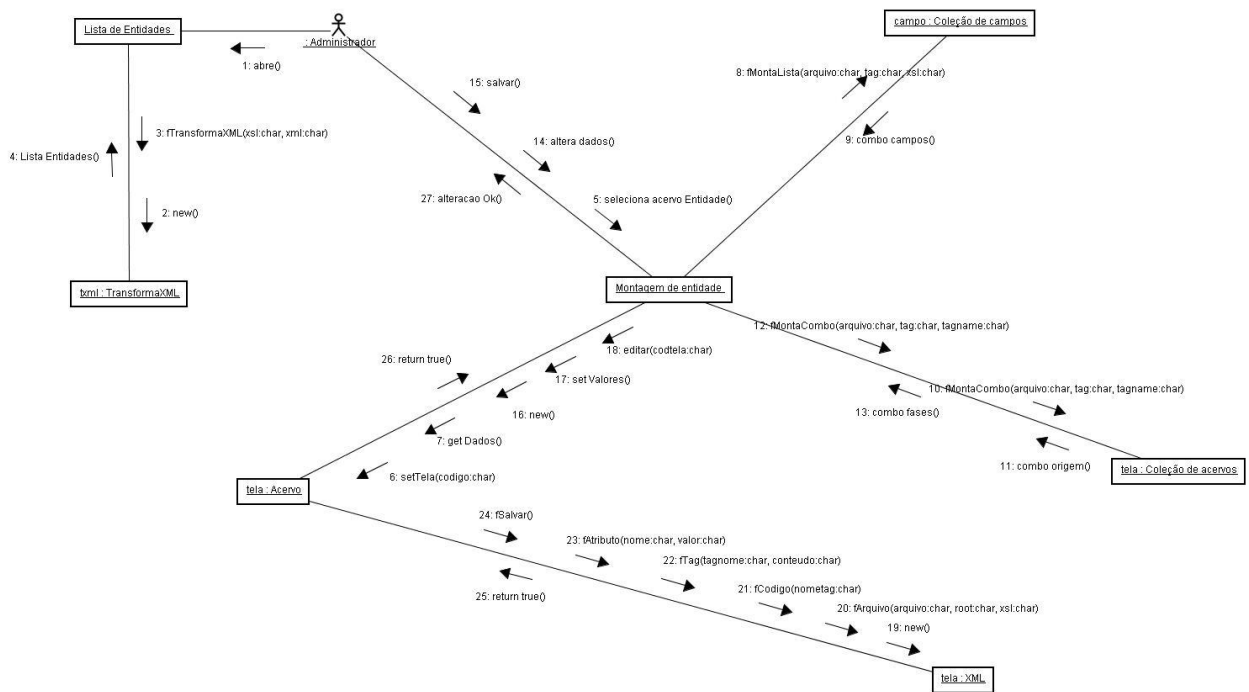


Figura 28 – Colaboração excluir Entidade

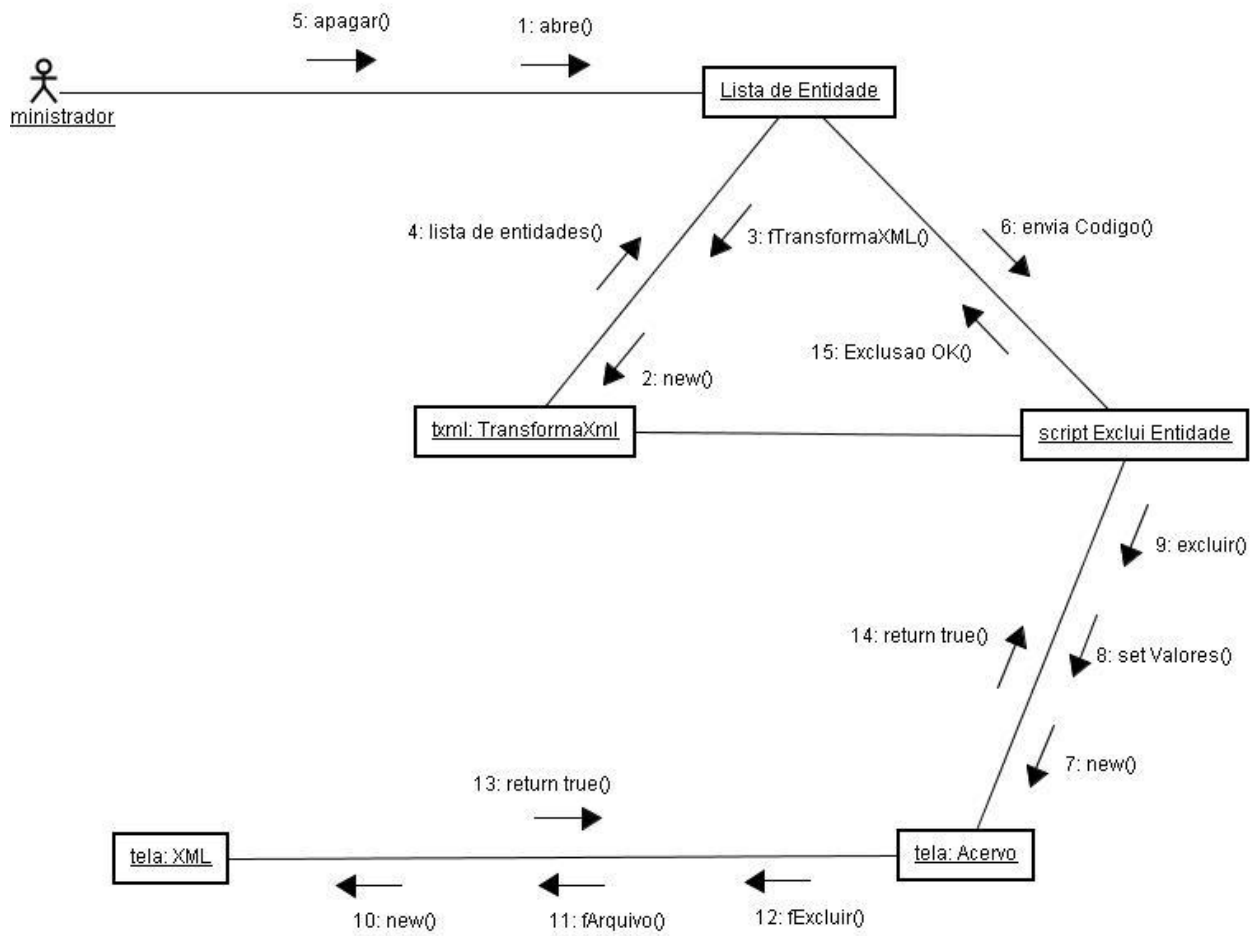


Figura 29 – Lista de Origens

Lista de Origens		
Nova		
Origem	Editar	Apagar

Figura 30 – Tela montar Origem

MONTAGEM DE ORIGEM		Voltar
TIPO DA ORIGEM:	<input type="text"/>	<input type="button" value="Salvar"/>
Campo:	<input type="text" value="Observações"/> <input type="button" value="Adicionar"/>	
Ordem:	<input type="text" value="3"/>	
Campos		
Período	<input type="text" value="Diurno"/>	<input type="button" value="Escolher"/>
Observações	<input type="text"/>	<input type="button" value="Escolher"/>

Figura 31 – Seqüência Incluir Origem

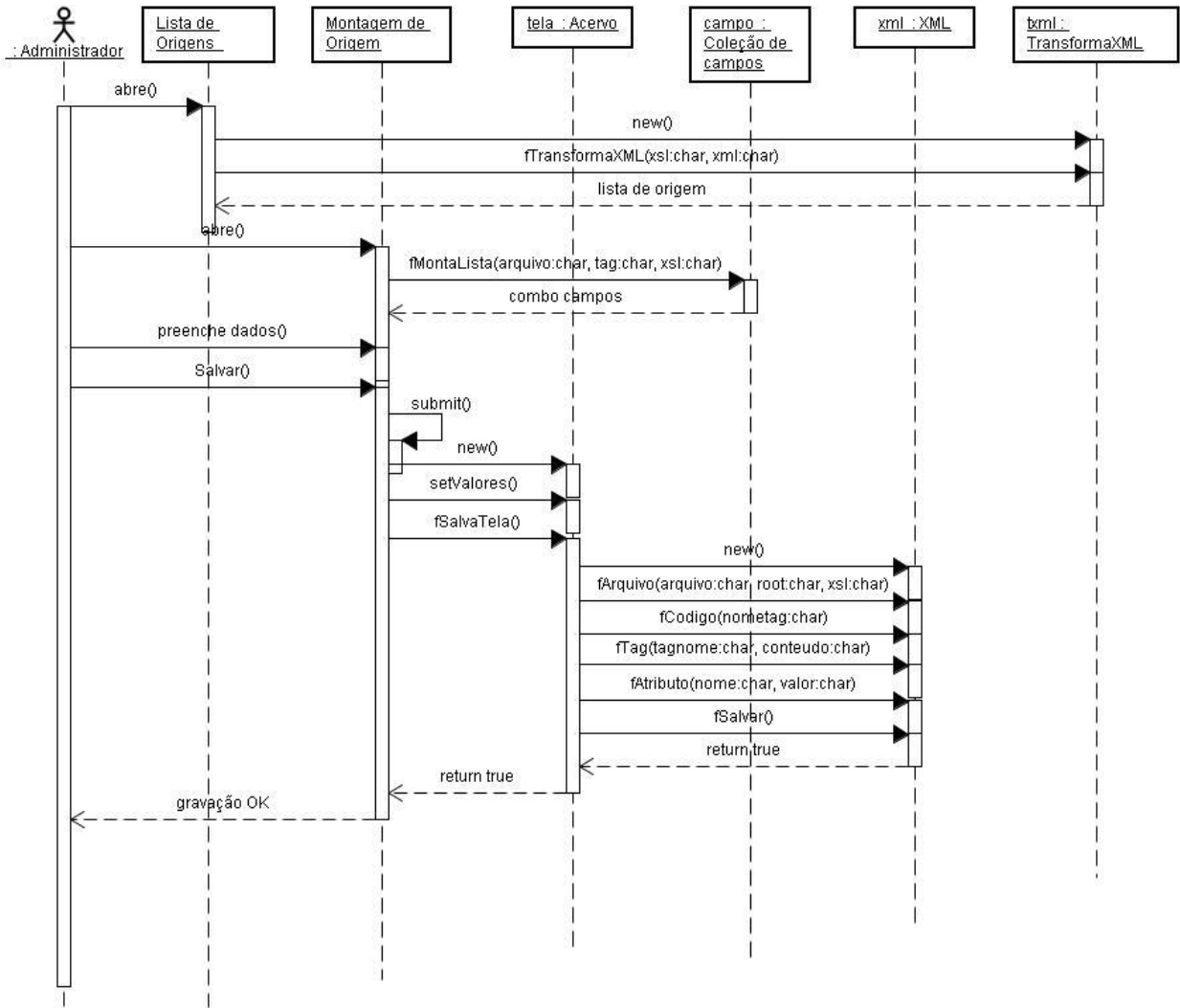


Figura 32 – Seqüência alterar Origem

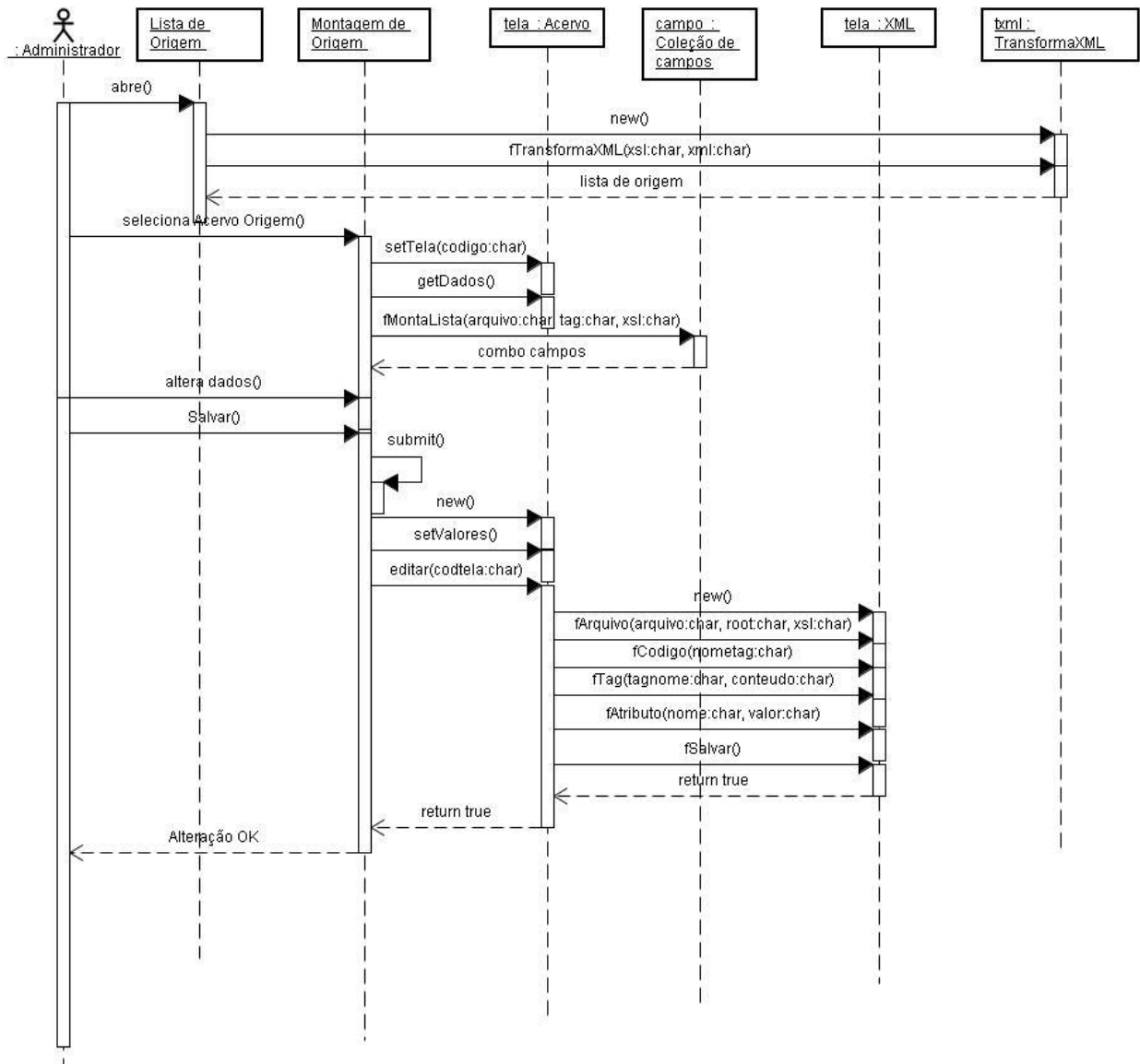


Figura 33 – Seqüência Excluir Origem

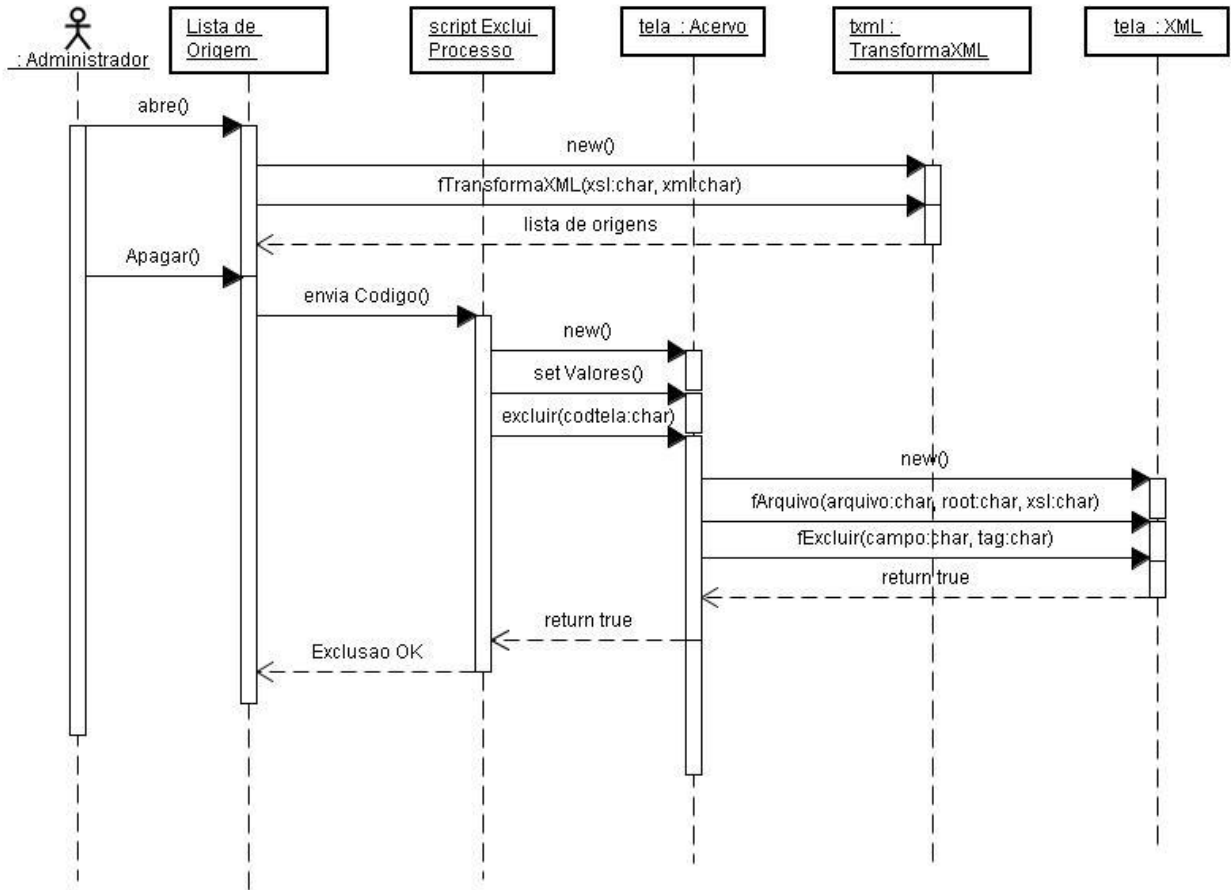


Figura 34 – Colaboração Incluir Origem

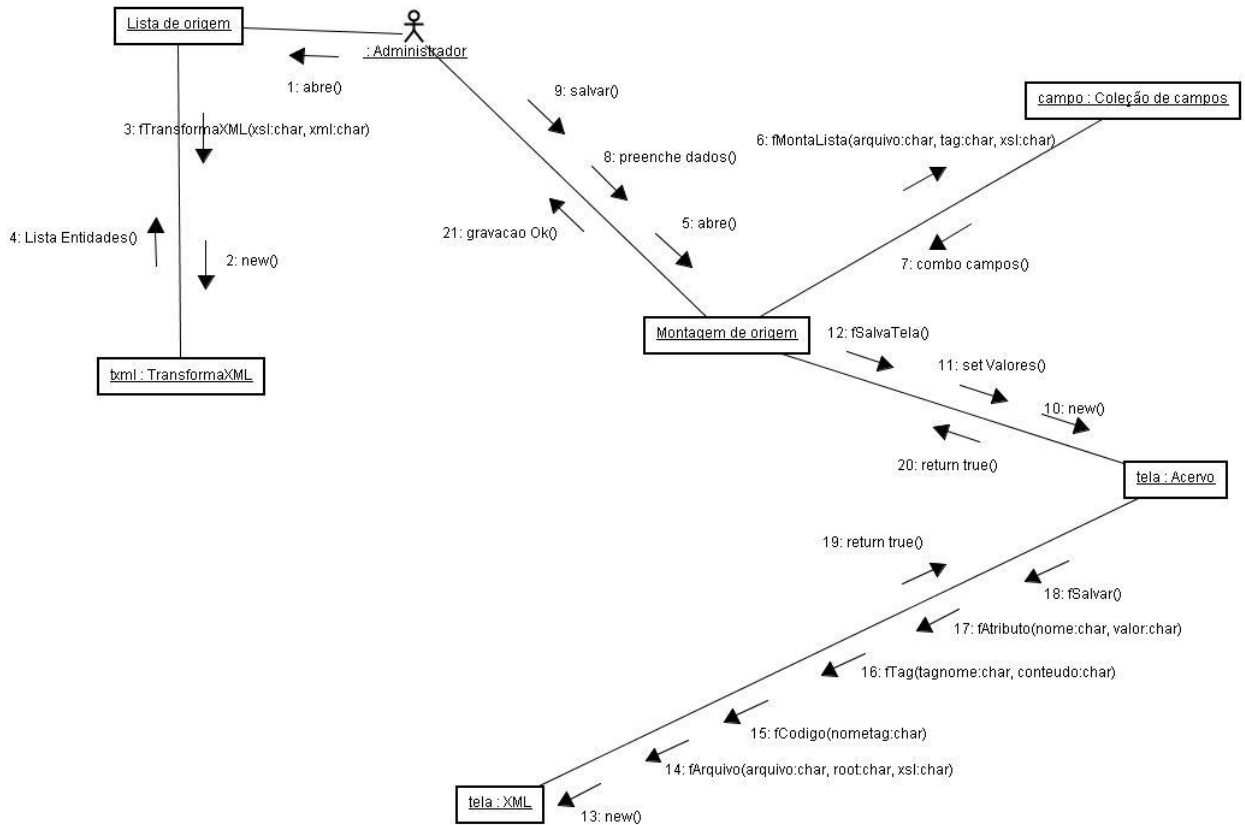


Figura 35 – Seqüência alterar Origem

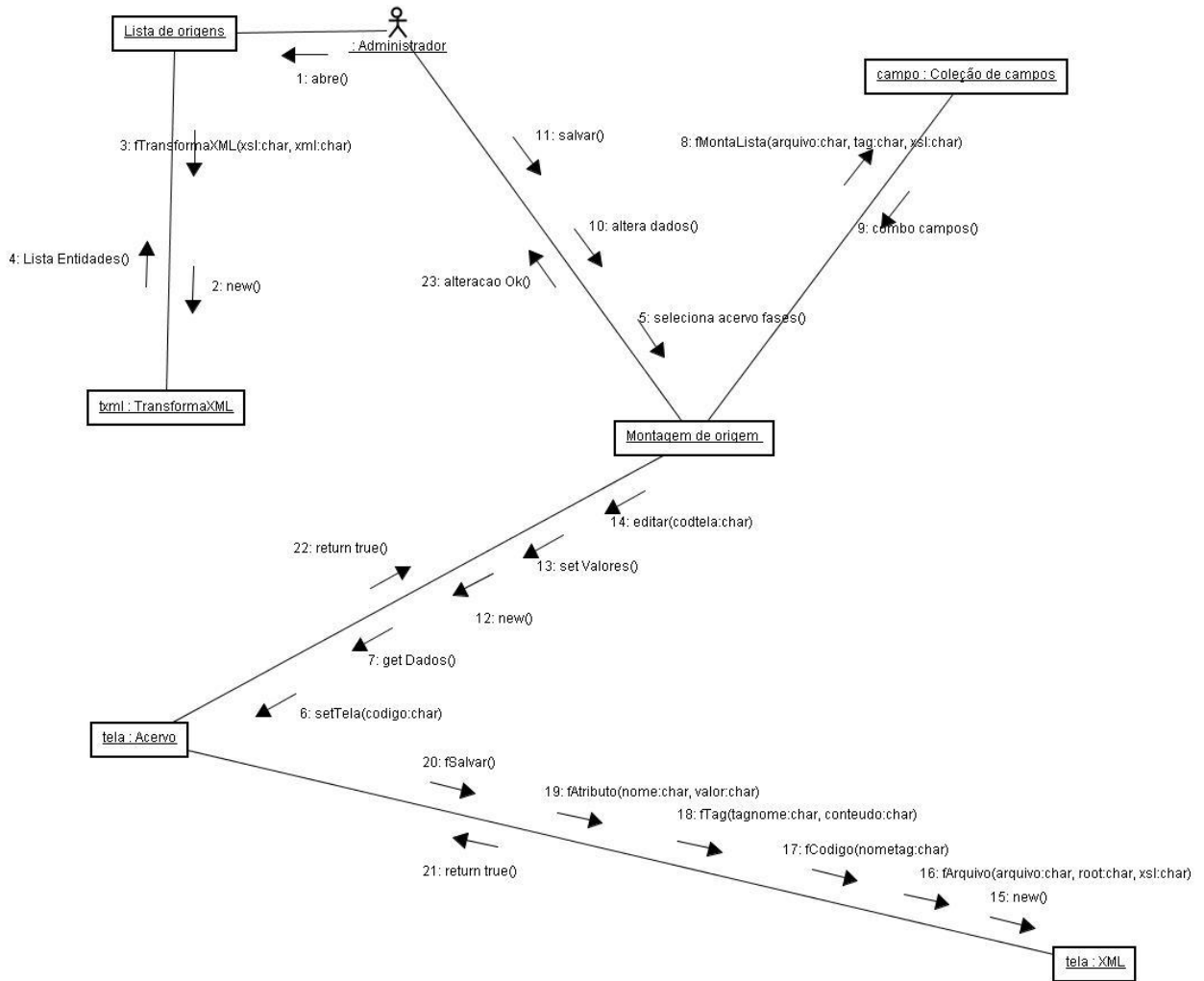


Figura 36 – Colaboração Excluir Origem

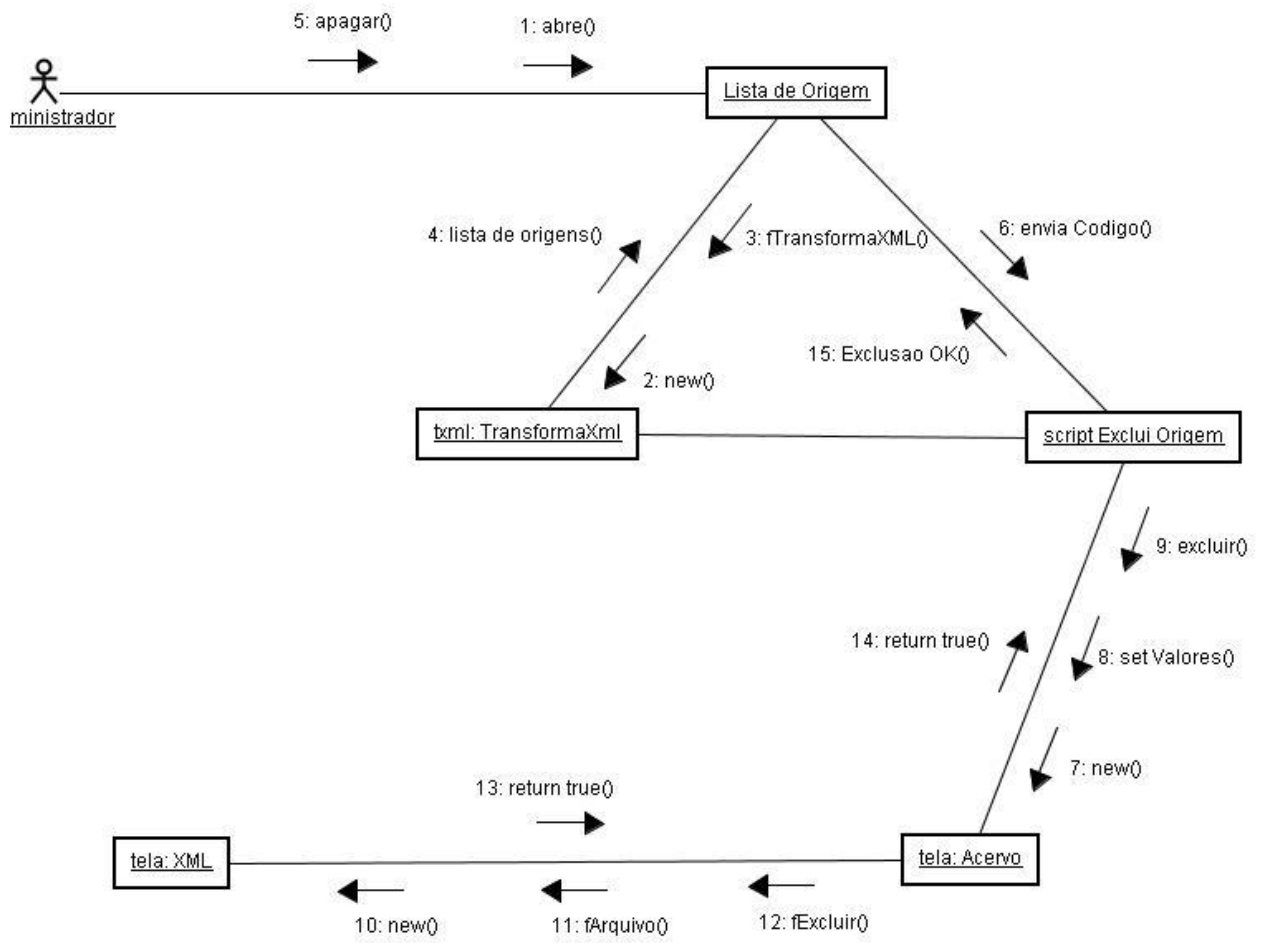


Figura 37 – Lista de fases

Lista de Fases		
Nova		
Recebimento	Editar	Apagar
Armazenamento	Editar	Apagar
Secagem	Editar	Apagar

Figura 38 – Tela Montar fase

MOTAGEM DE FASE Voltar

NOME DA FASE:

Campo: Ordem:

Campos

Período	Diurno	Excluir

Lista de Processos

Plantio	Excluir
Adubação	Excluir

Figura 39 – Seqüência Cadastrar Fases

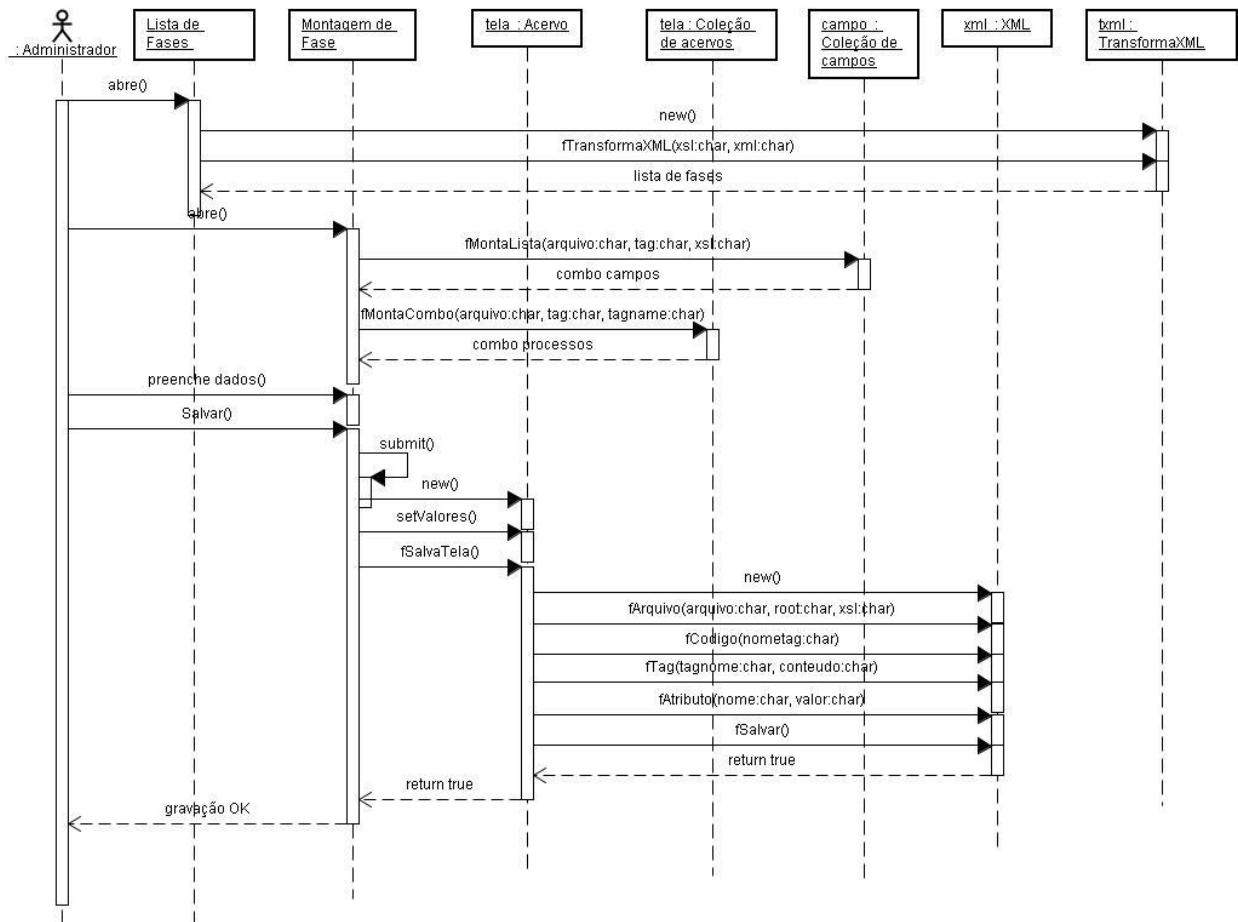


Figura 40 – Seqüência Alterar Fases

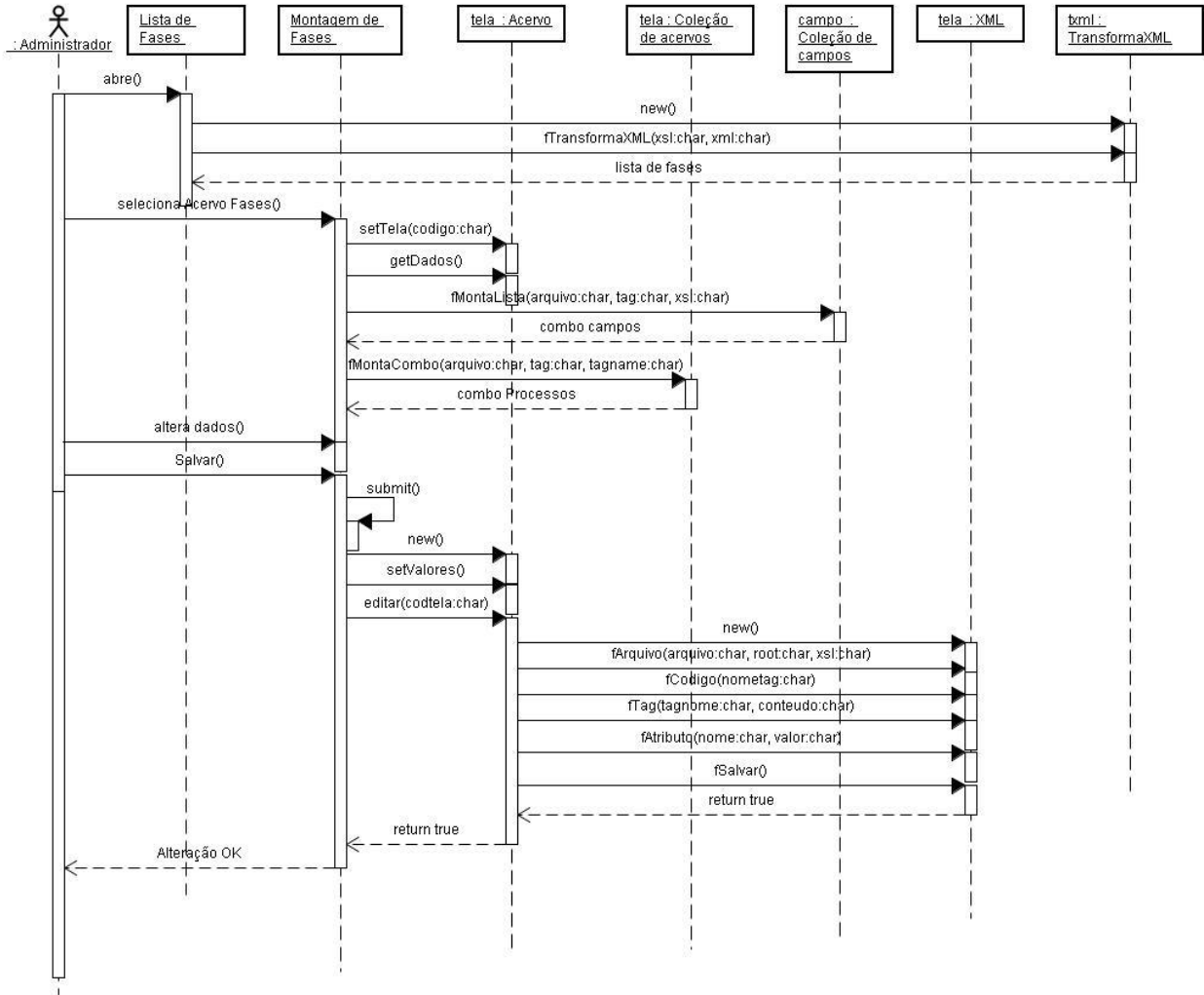


Figura 41 – Seqüência excluir Fase

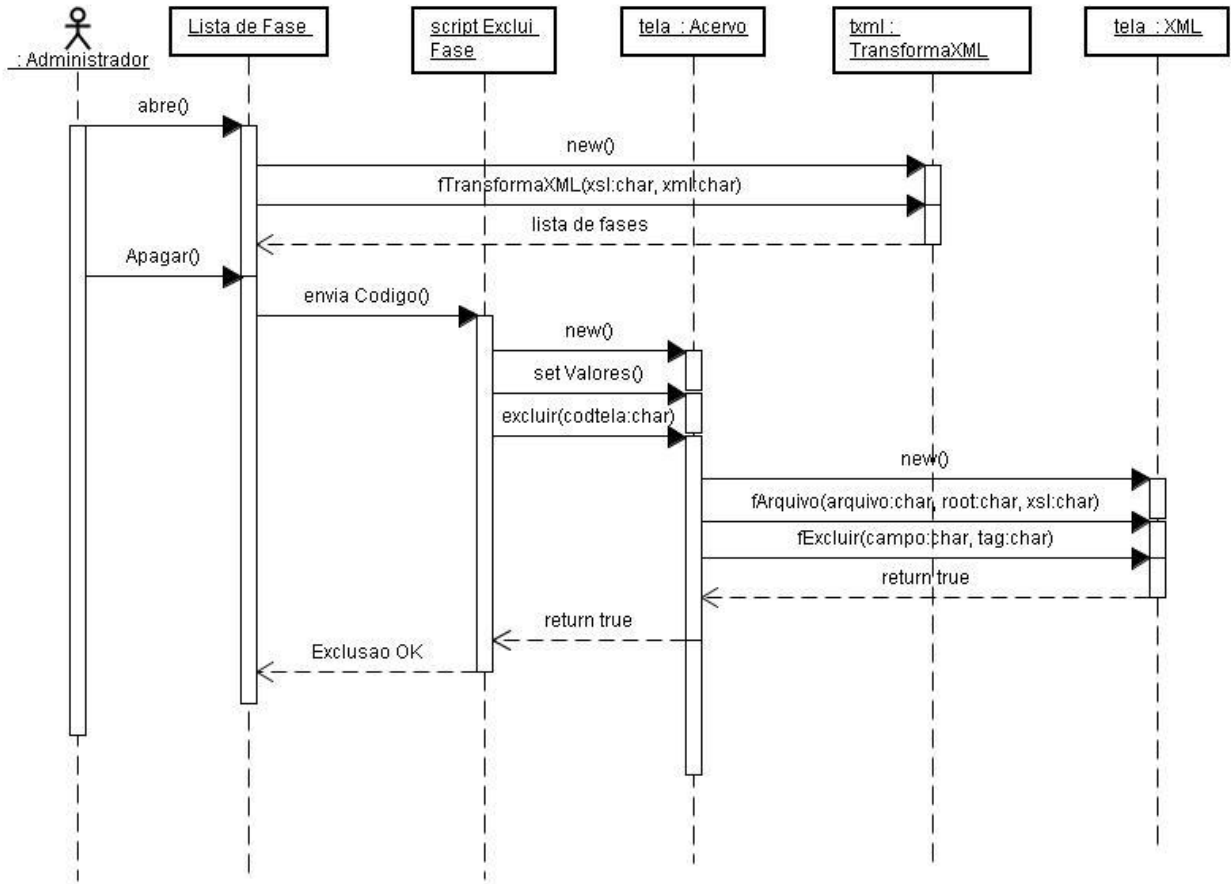


Figura 42 – Colaboração Cadastrar Fases

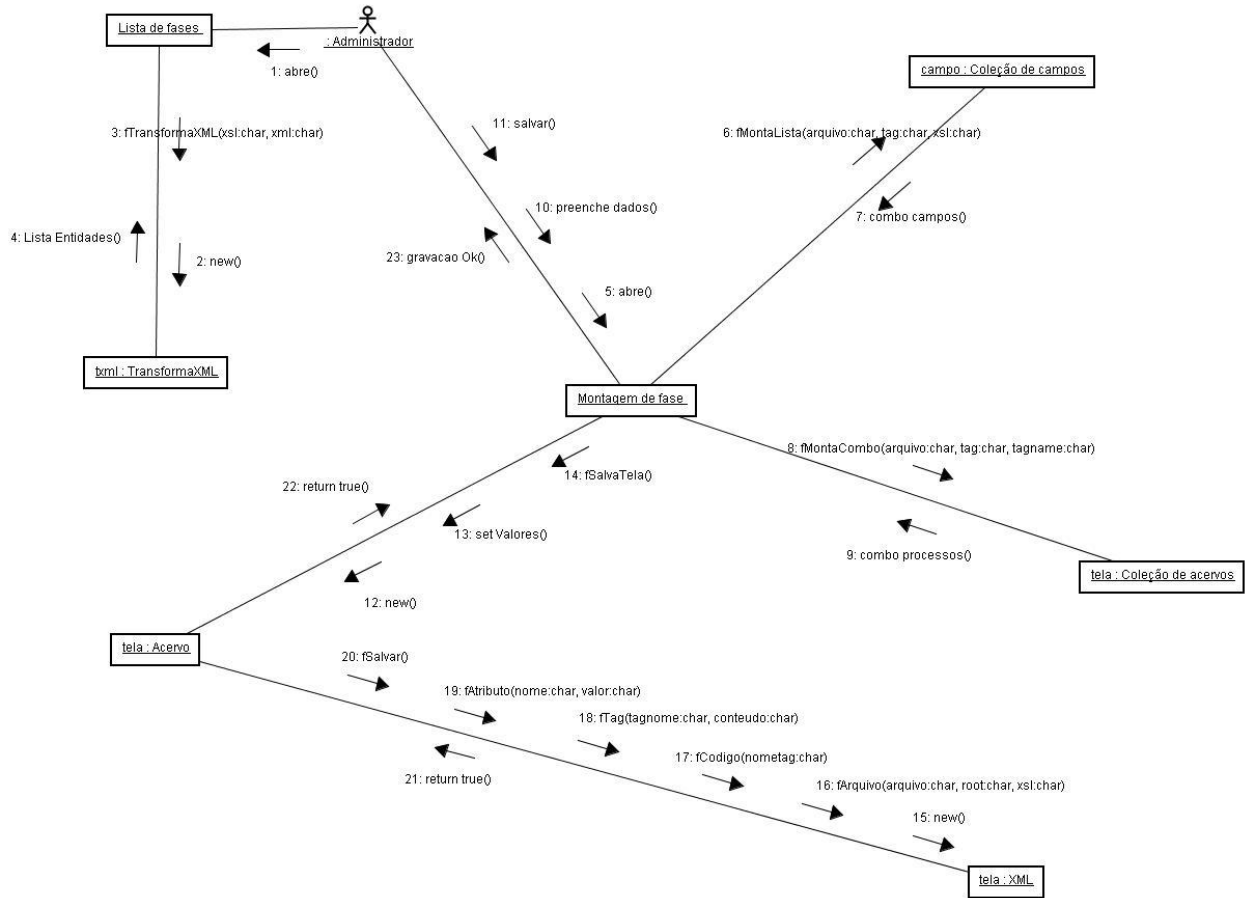


Figura 43 – Colaboração Alterar Fases

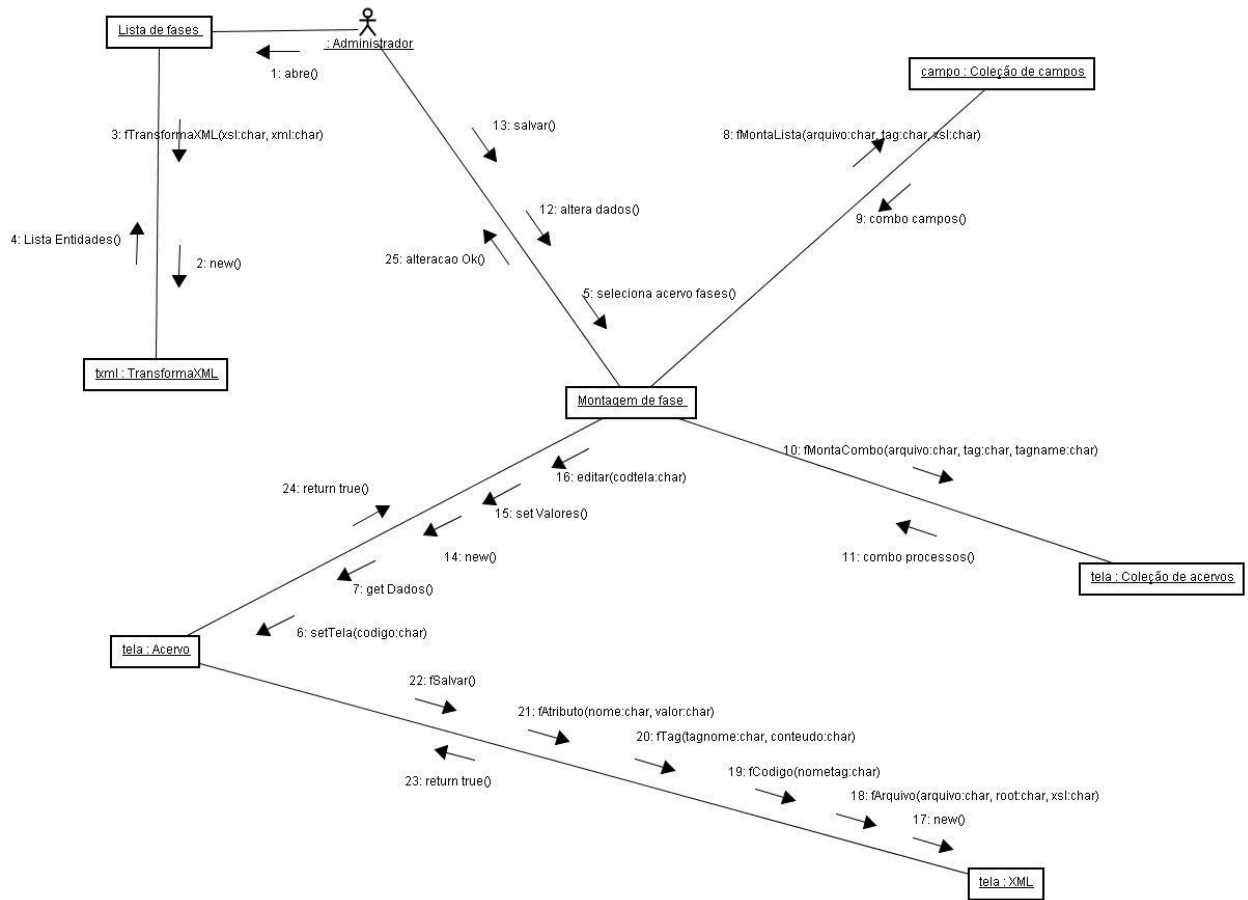


Figura 44 – Colaboração excluir Fase

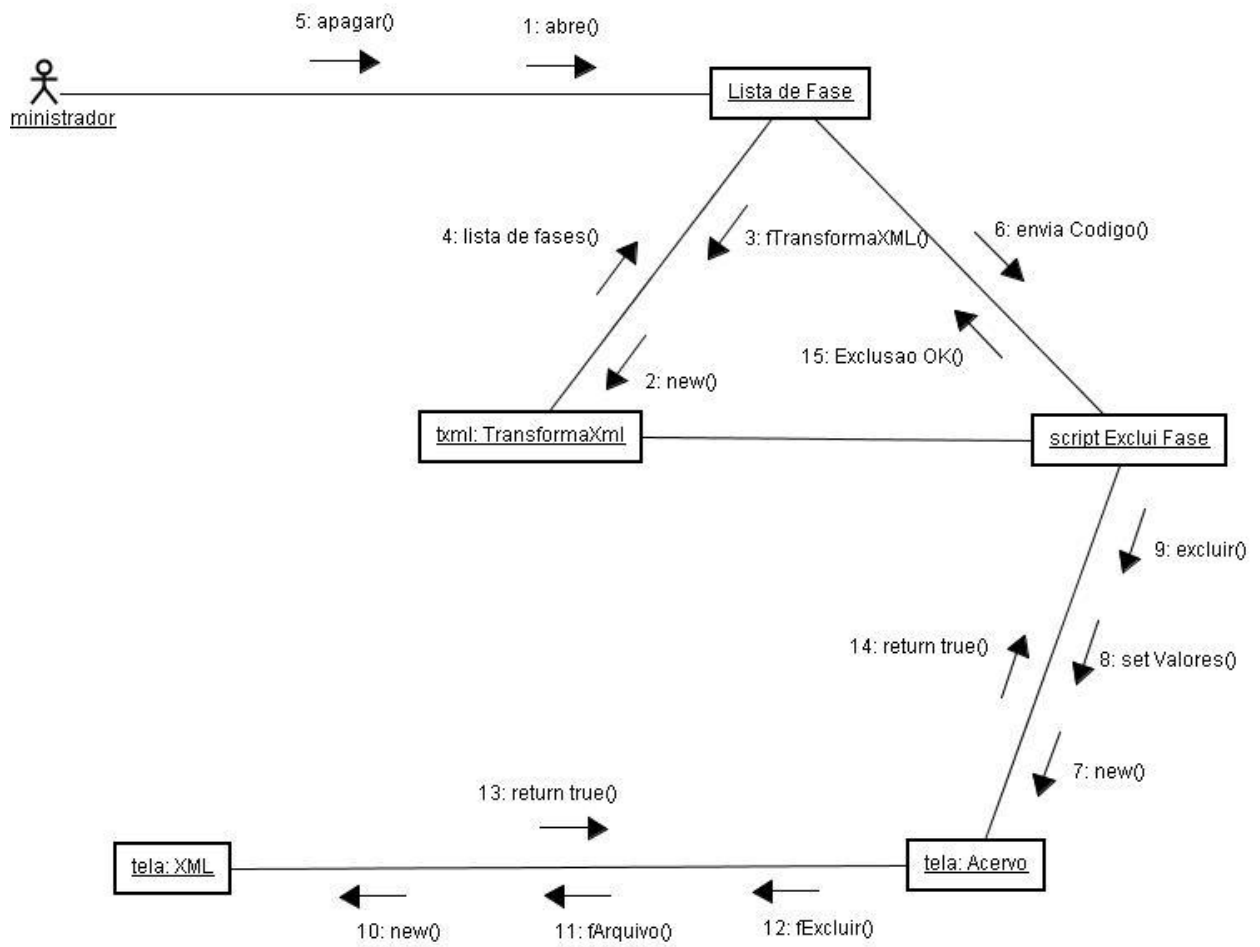


Figura 45 – Lista de processos

Lista de Processos		
Novo		
Monitoramento	Editar	Apagar
Termometria	Editar	Apagar
Resfriamento	Editar	Apagar
Aplicação de inseticida	Editar	Apagar
Aplicação de Terra de Diatomácea	Editar	Apagar
Amostragem/Cassificação	Editar	Apagar

Figura 46 – Tela montar processo

MOTAGEM DE PROCESSO		Voltar
NOME DO PROCESSO:	<input type="text" value="Adubação"/>	<input type="button" value="Salvar"/>
Campo:	<input type="text" value="Nome"/> <input type="button" value="Adicionar"/>	Ordem: <input type="text" value="3"/>
Campos		
Produtos Químicos	<input type="text" value="Herbicida"/>	<input type="button" value="Excluir"/>
Nome	<input type="text" value="Eduardo"/>	<input type="button" value="Excluir"/>

Figura 47 – Seqüência cadastrar Processo

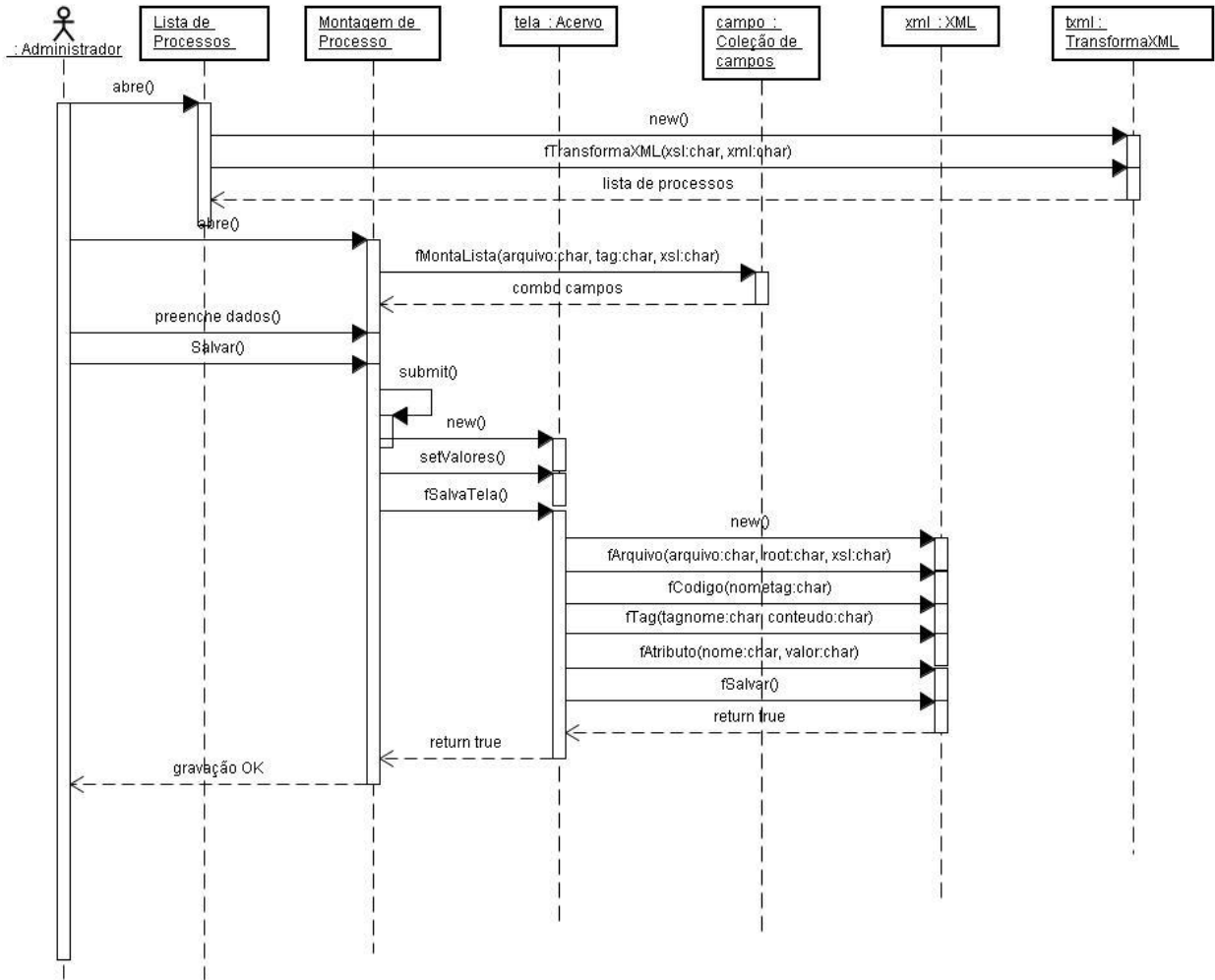


Figura 48 - Seqüência alterar processo

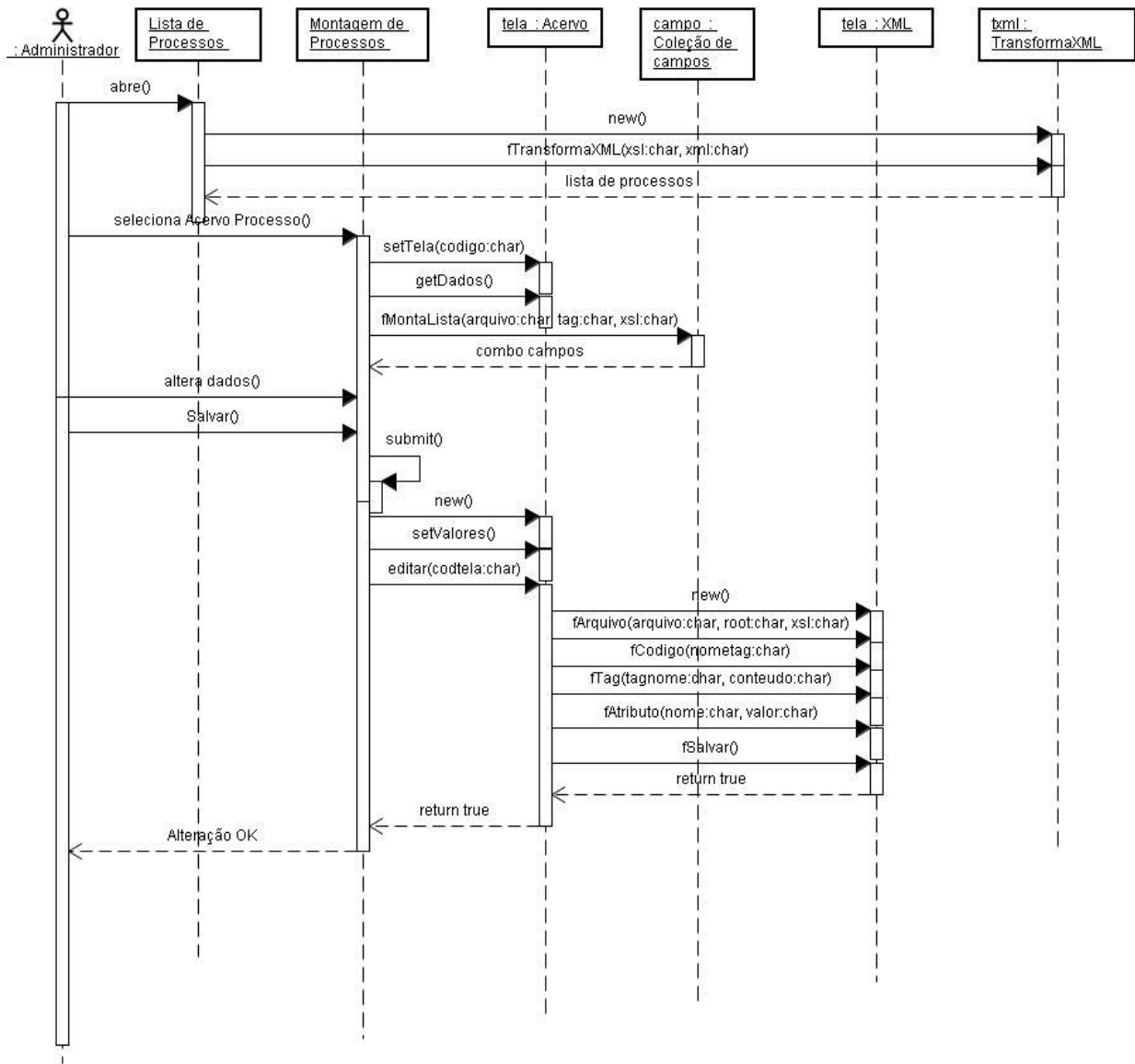


Figura 49 – Seqüência excluir processo

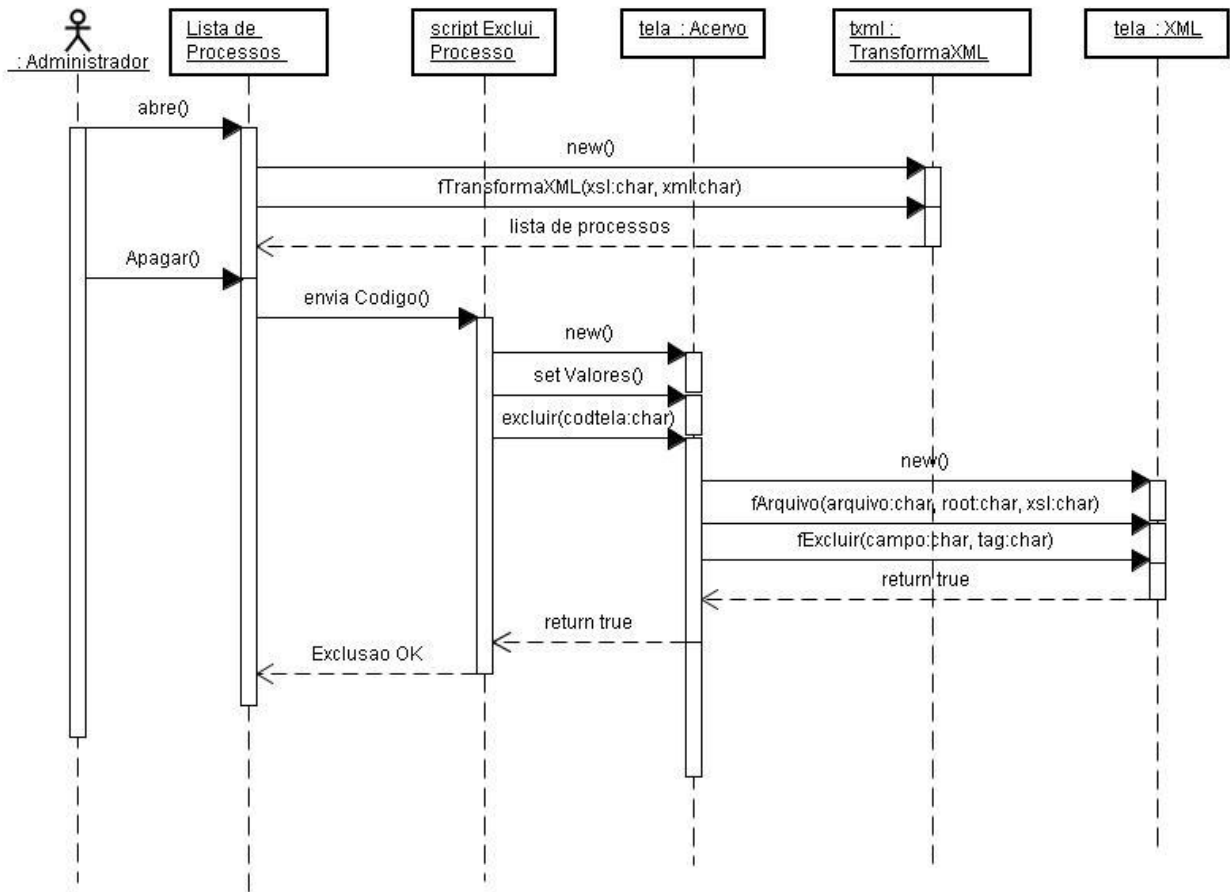


Figura 50 – Colaboração cadastrar Processo

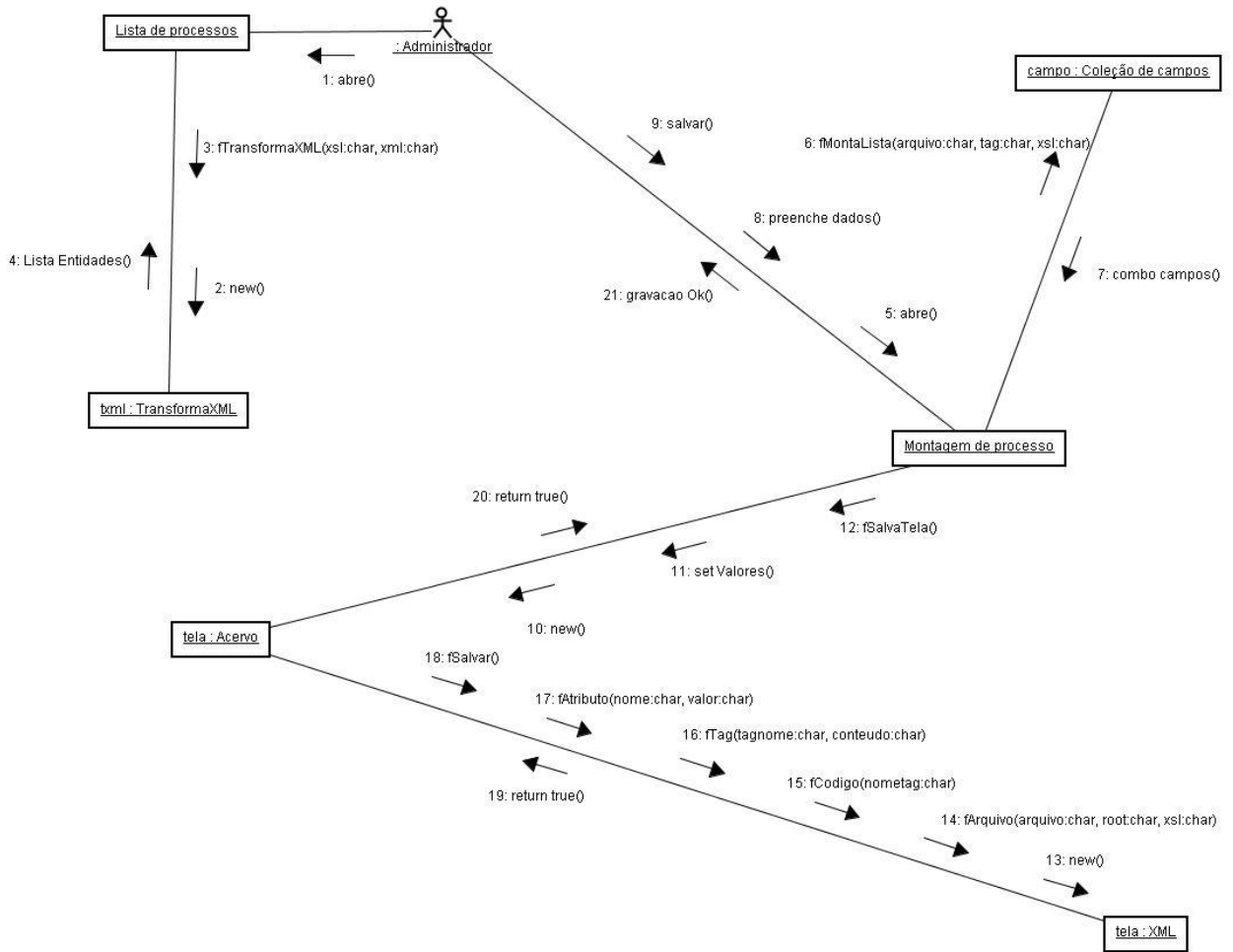


Figura 51 - Colaboração alterar processo

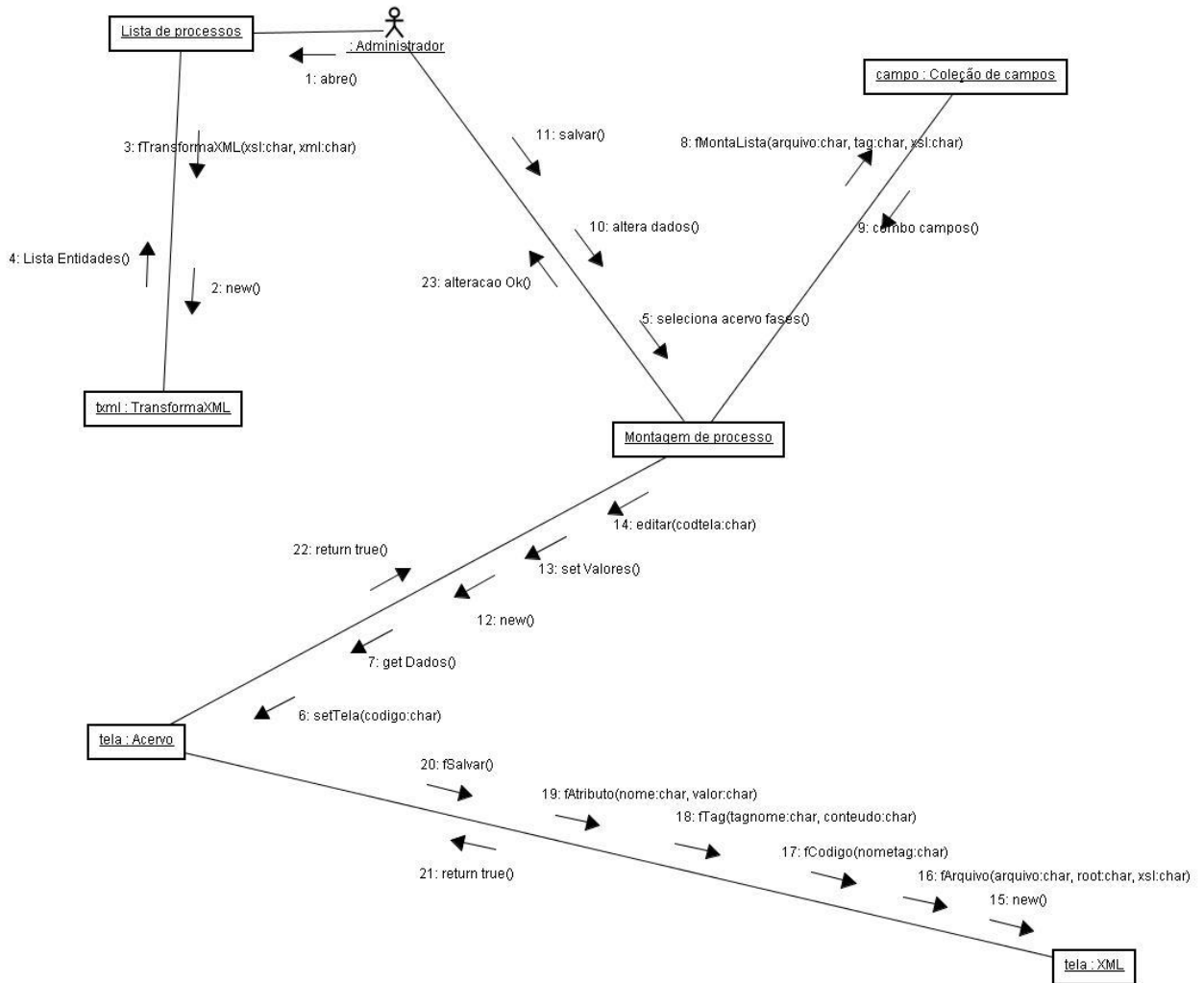


Figura 52 – Colaboração excluir processo

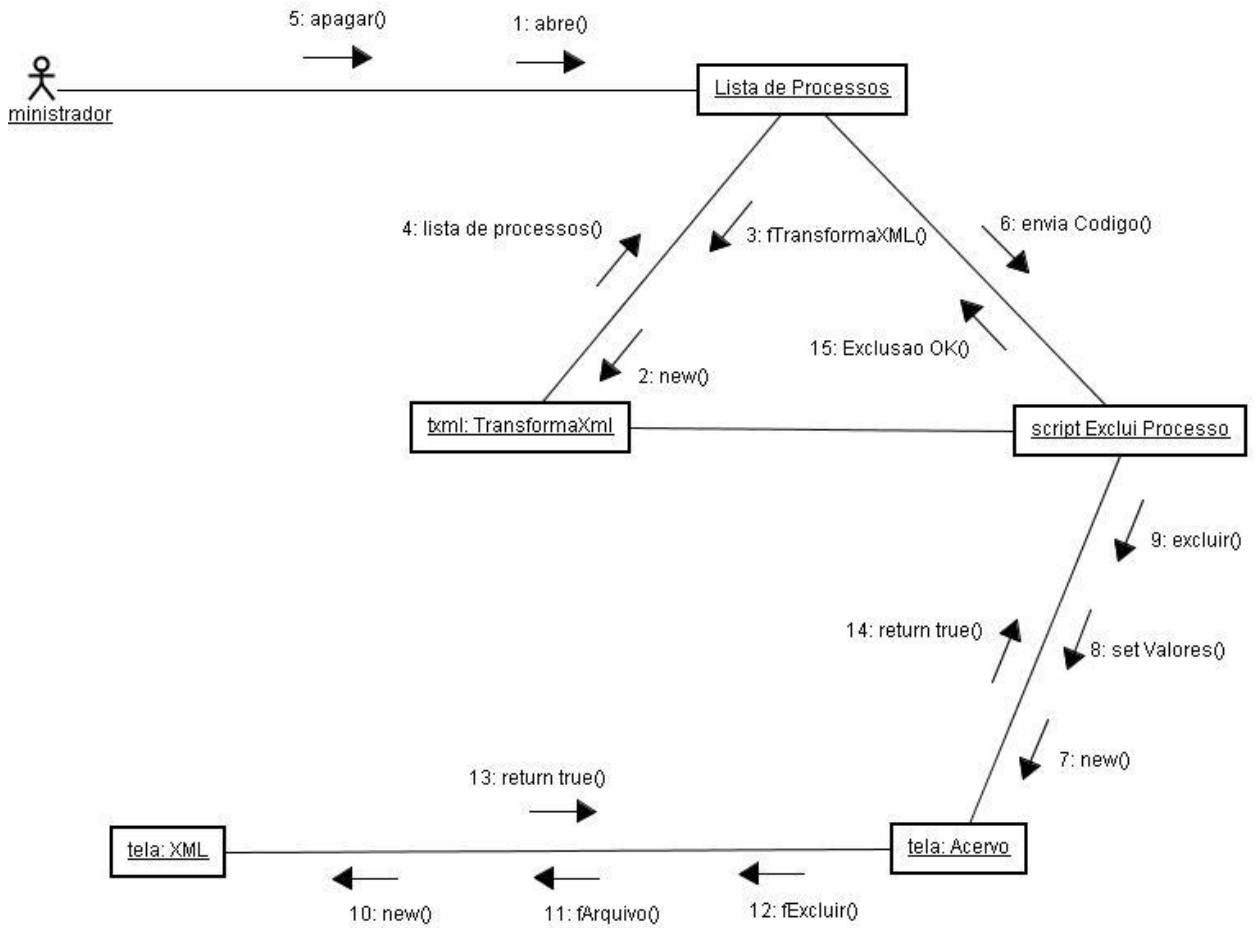


Figura 53 – Tela alterar senha



Figura 54 – Seqüência Alterar Senha

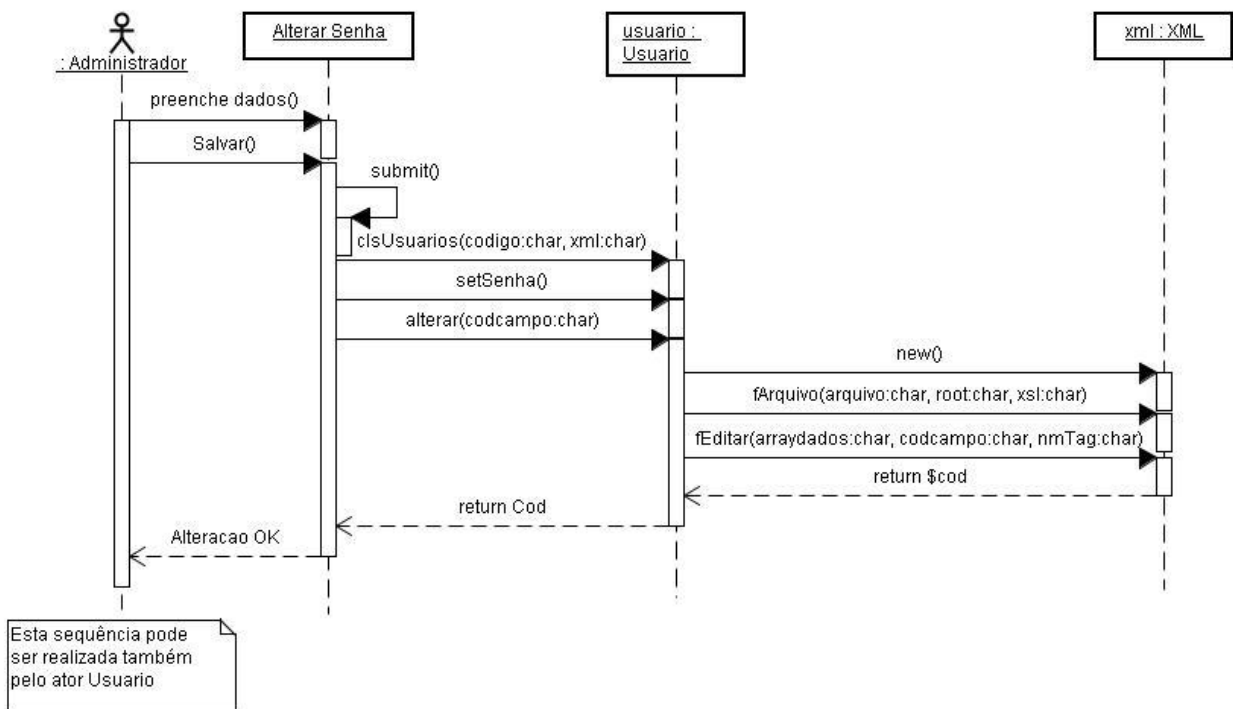


Figura 55 – Colaboração Alterar Senha

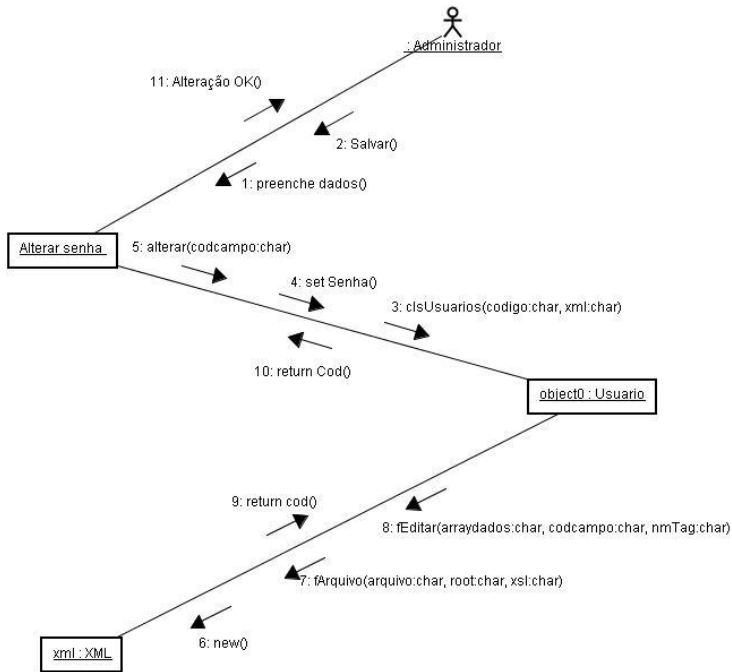


Figura 56 – Registro de entidade

ENTIDADE - Milho nº: 5

[RELATÓRIO COMPLETO](#) | [EDITAR](#) | [EXCLUIR](#)

[Agrupar](#) | [Desagrupar](#)

Semente:

Produtos Químicos: Herbicida

Ocorrências: Não houveram ocorrências

Observações:

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 57 - Tela Agrupar entidades

Agrupar - Milho 6

Entidades:

Milho - 4

Milho - 5

Agrupamento:

Milho - 3
Milho - 5

[Agrupar](#) [Cancelar](#)

Webmaster: Ana Maria Pereira e Eduardo Rafael Coutinho dos Santos

Figura 58 – Seqüência agrupar entidades

A sequencia Incluir Registro de entidade deve ser executada antes do Agrupamento

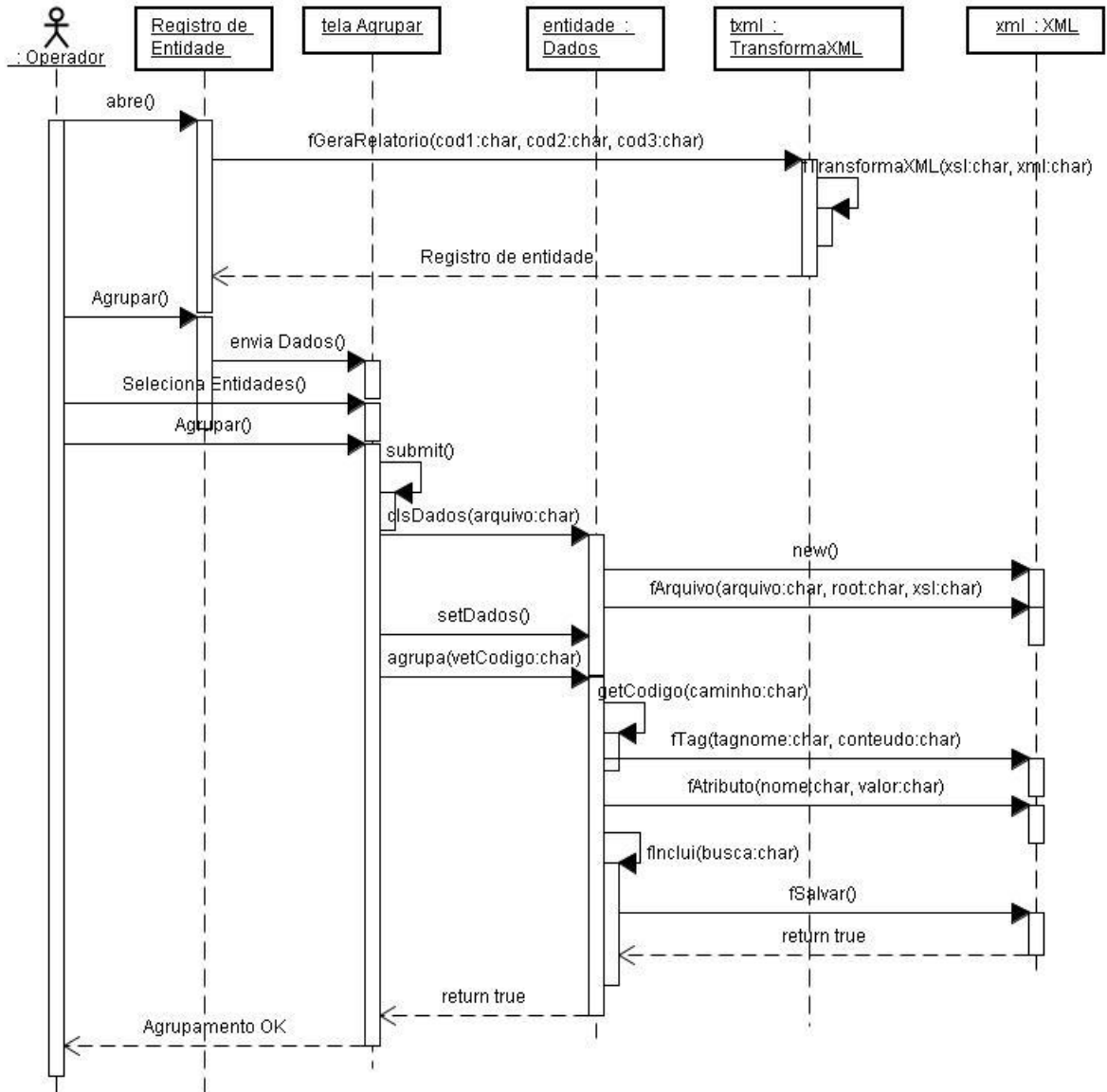


Figura 59 – Seqüência desagrupar entidade

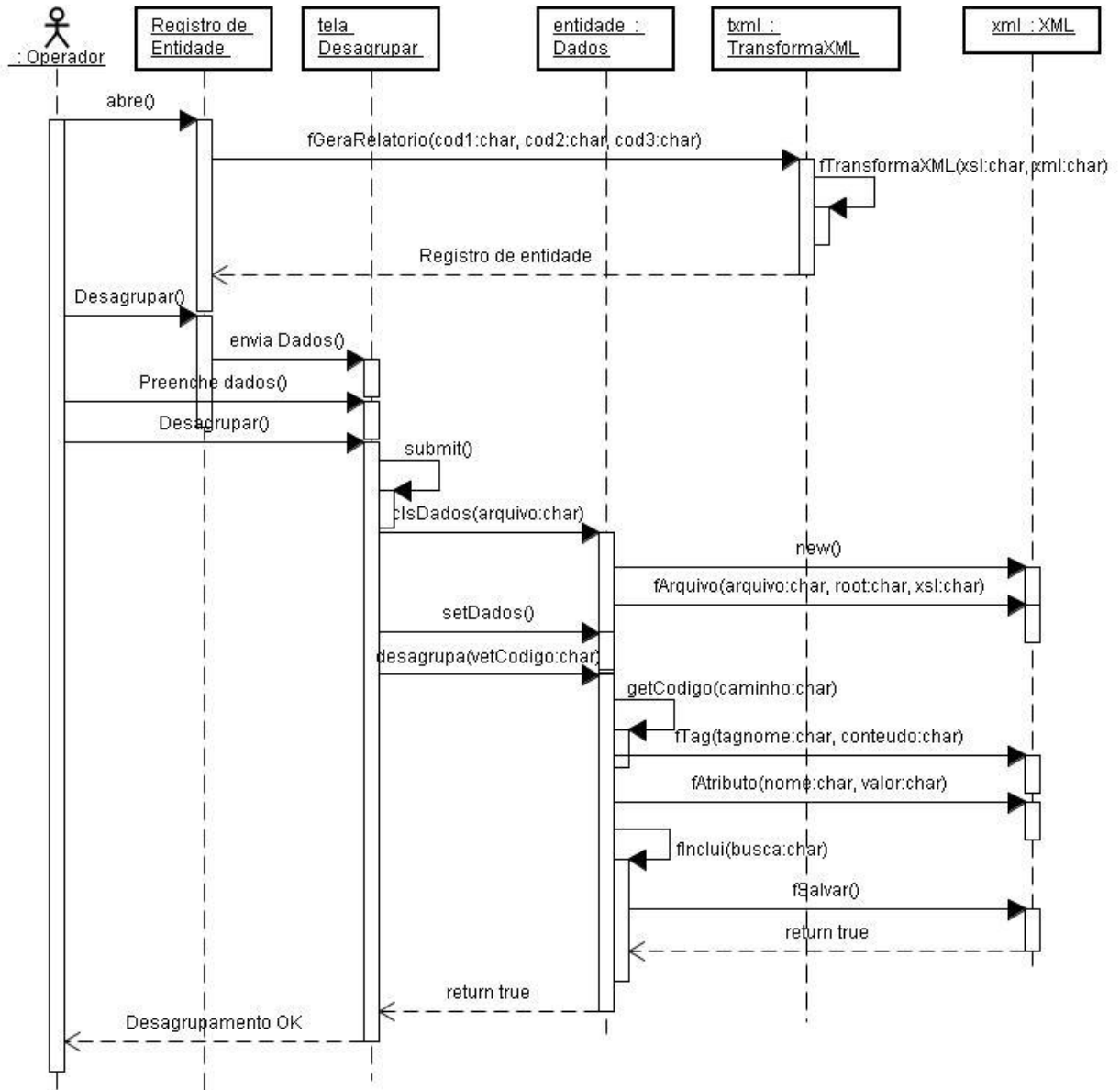


Figura 60 – Colaboração agrupar entidades

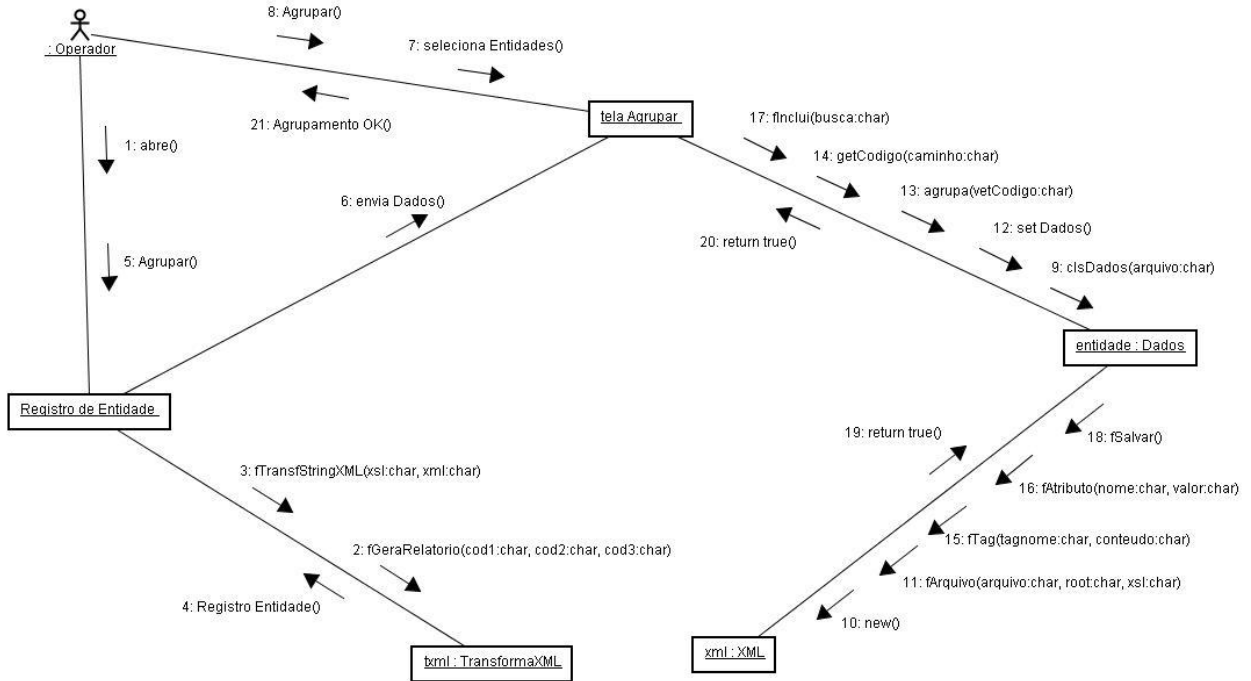


Figura 61 – Colaboração desagrupar entidade

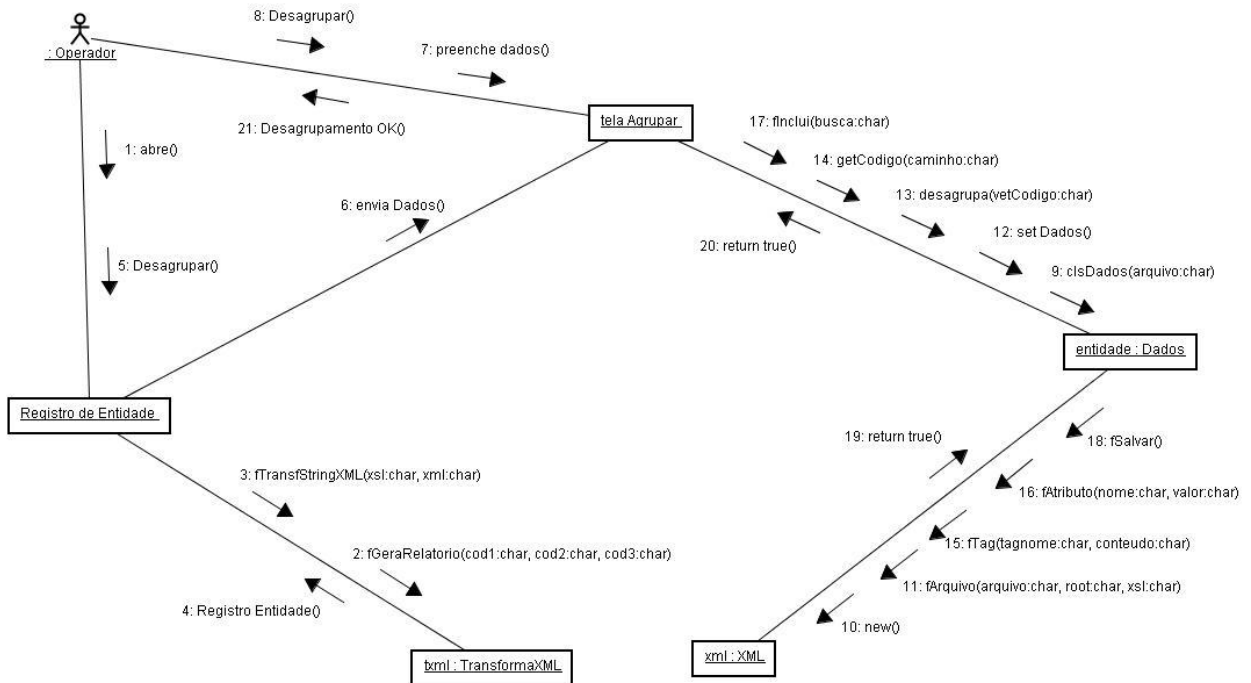


Figura 62 – Tela Incluir Registro de Entidade

Objeto da rastreabilidade: **Milho**

CADASTRO DE ENTIDADE - Milho

Peso:

Variedade:

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 63 – Seqüência Incluir Registro de Entidade

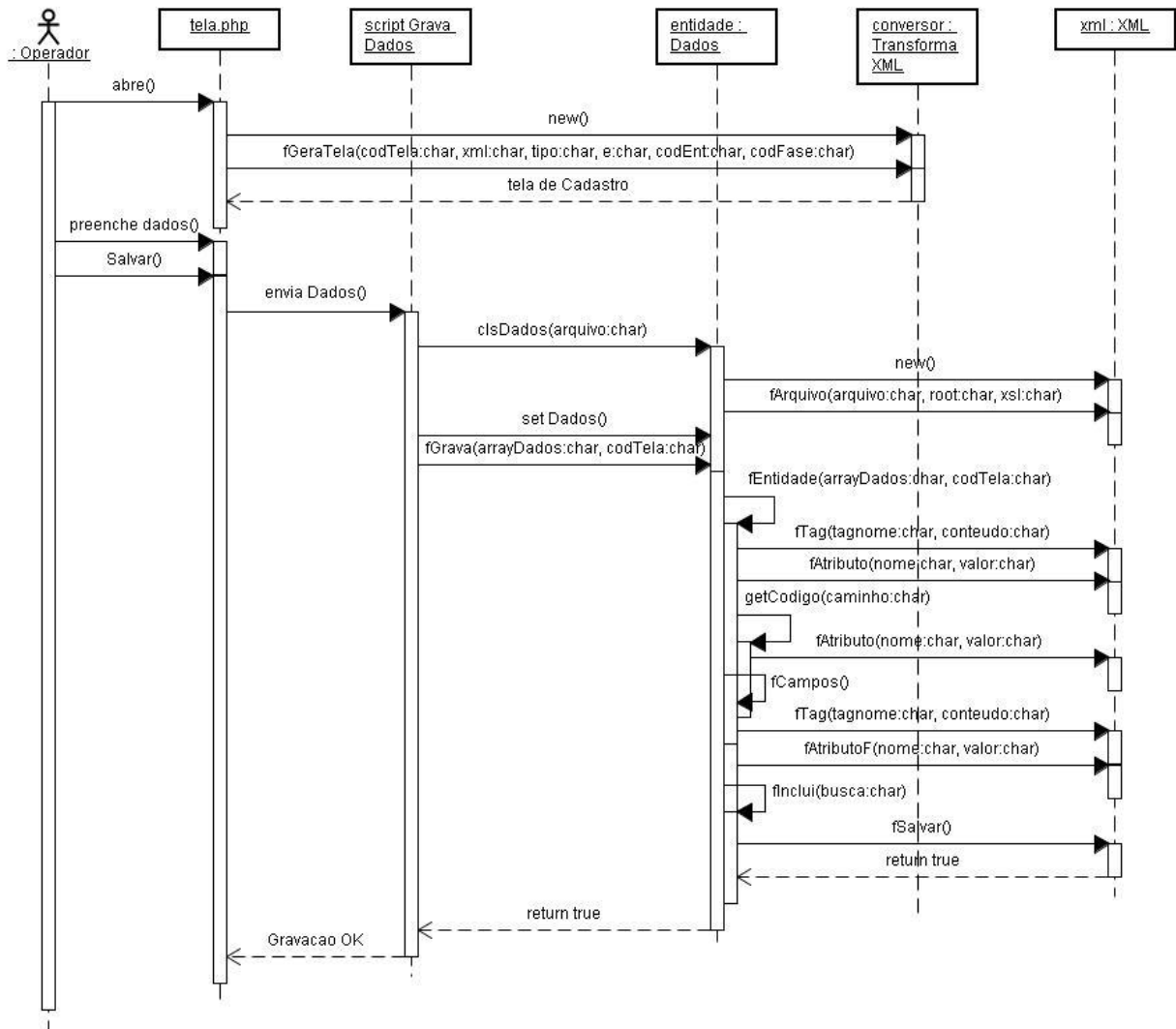


Figura 64 – Seqüência Alterar Registro de Entidade

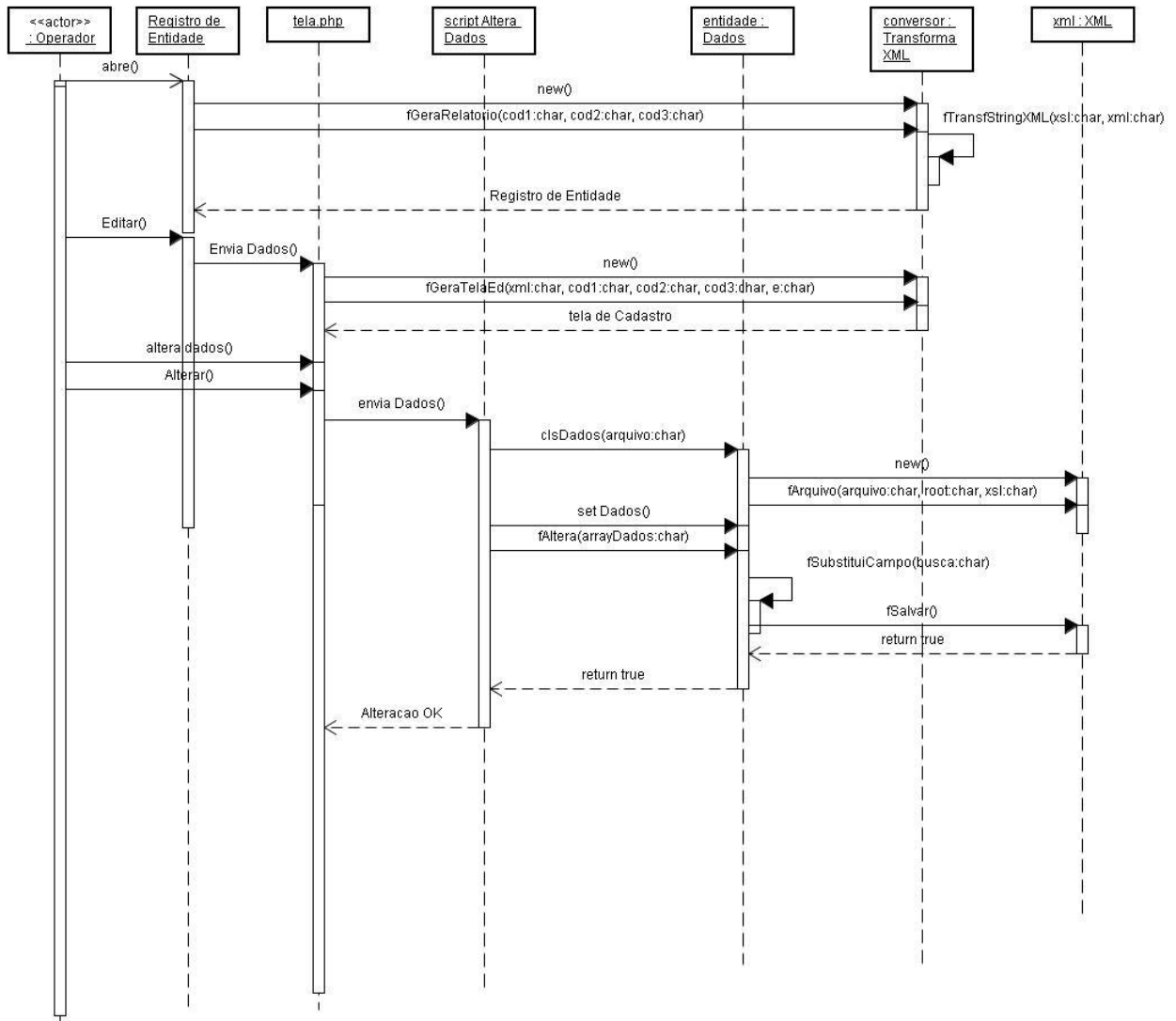


Figura 65 – Seqüência Excluir Registro de Entidade

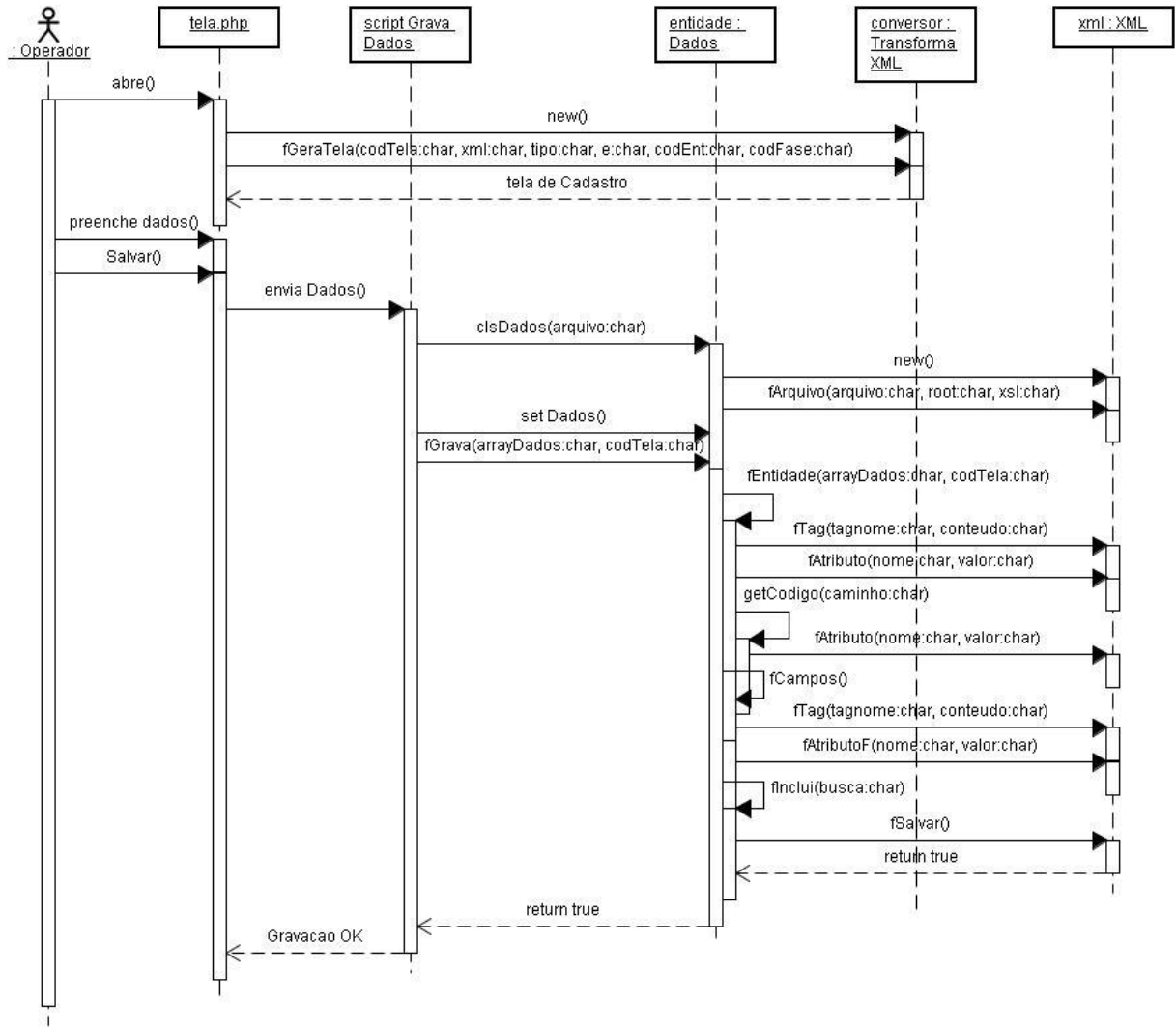


Figura 66 – Colaboração Incluir Registro de Entidade

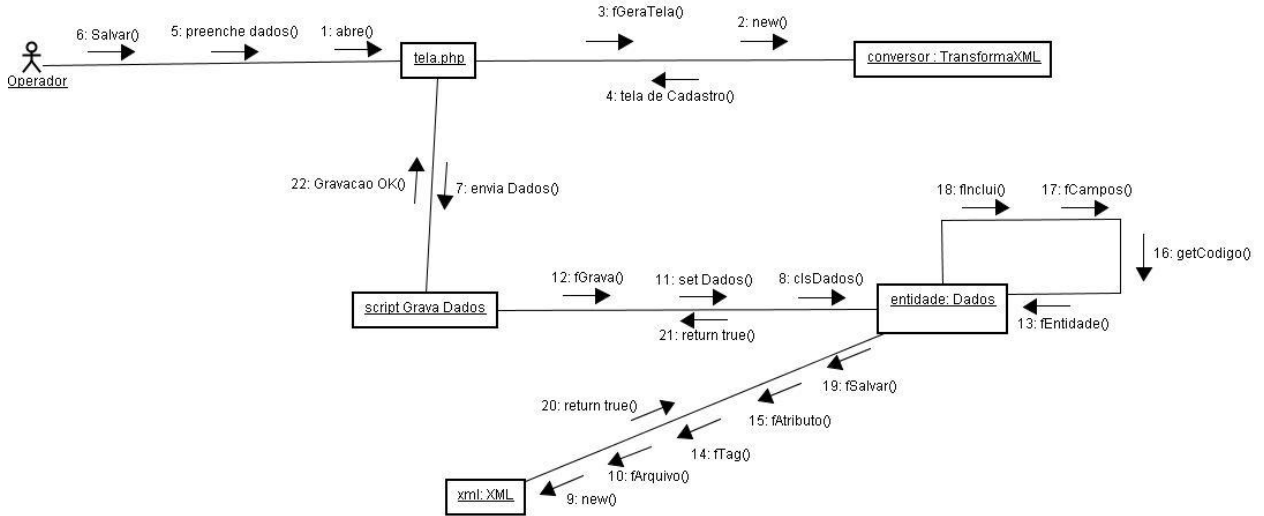


Figura 67 – Colaboração Alterar Registro de Entidade

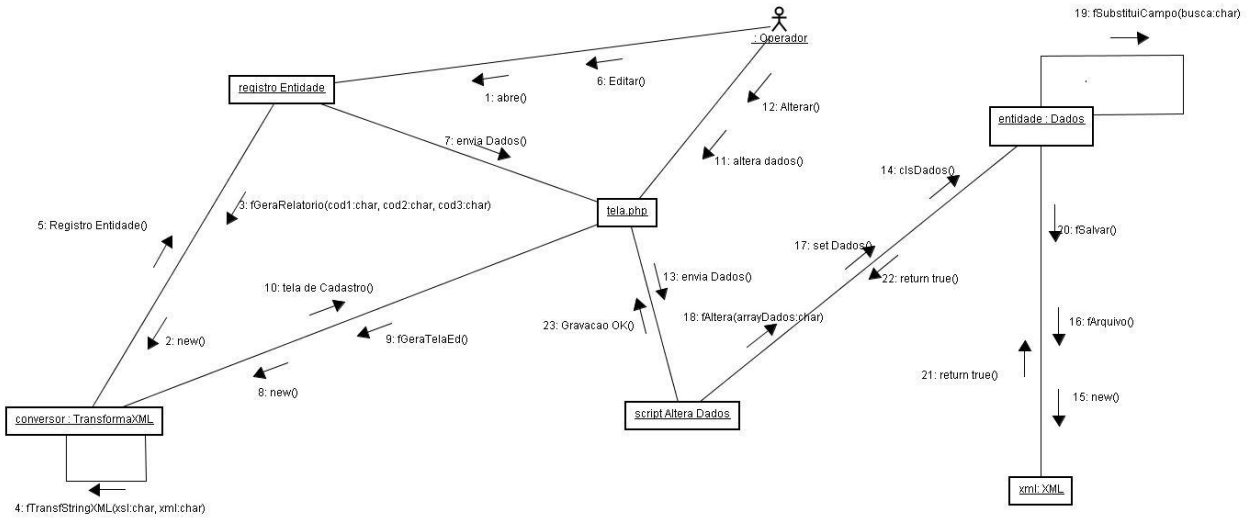


Figura 68 – Colaboração Excluir Registro de Entidade

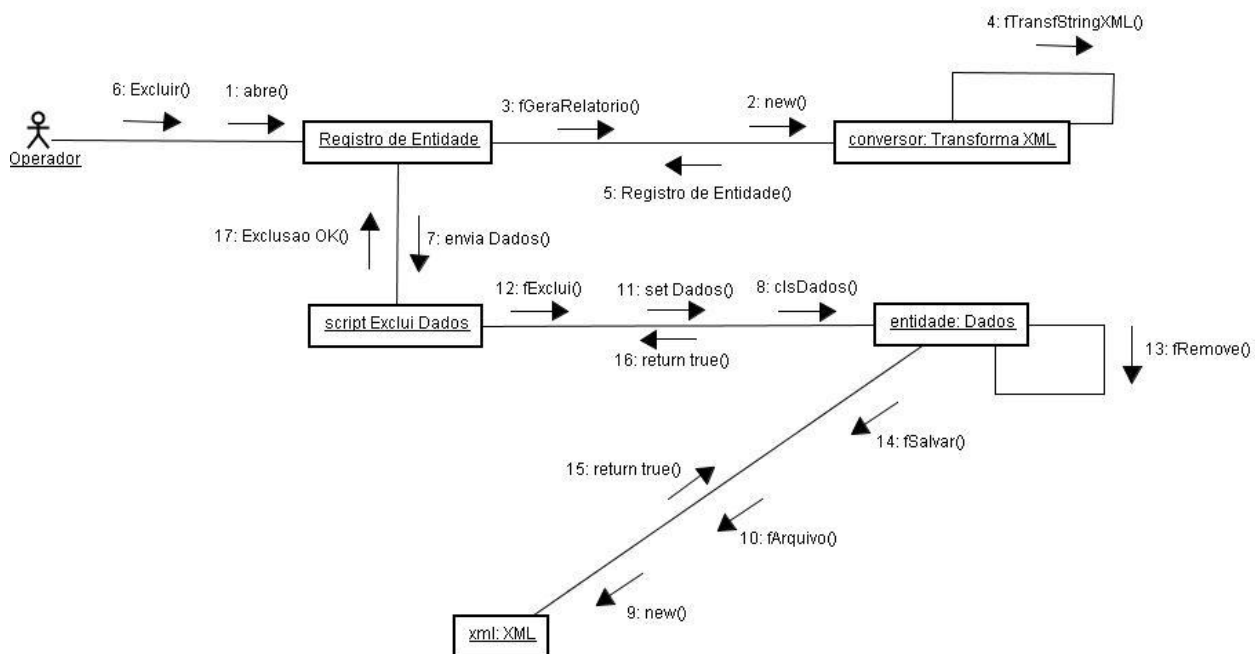


Figura 69 – Registro de Fase

Fase: Secagem	Milho nº: 6
	EDITAR
Quantidade: 500 kg	
Temperatura medida: 32°	

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 70 – Tela incluir Registro de Fase

CADASTRO DE FASE - Armazenamento	Milho nº: 4
Silo:	<input type="text"/>
Data de Entrada:	<input type="text"/>
Data de saída:	<input type="text"/>
<input type="button" value="Salvar"/>	

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 71 – Seqüência Incluir Registro de Fase

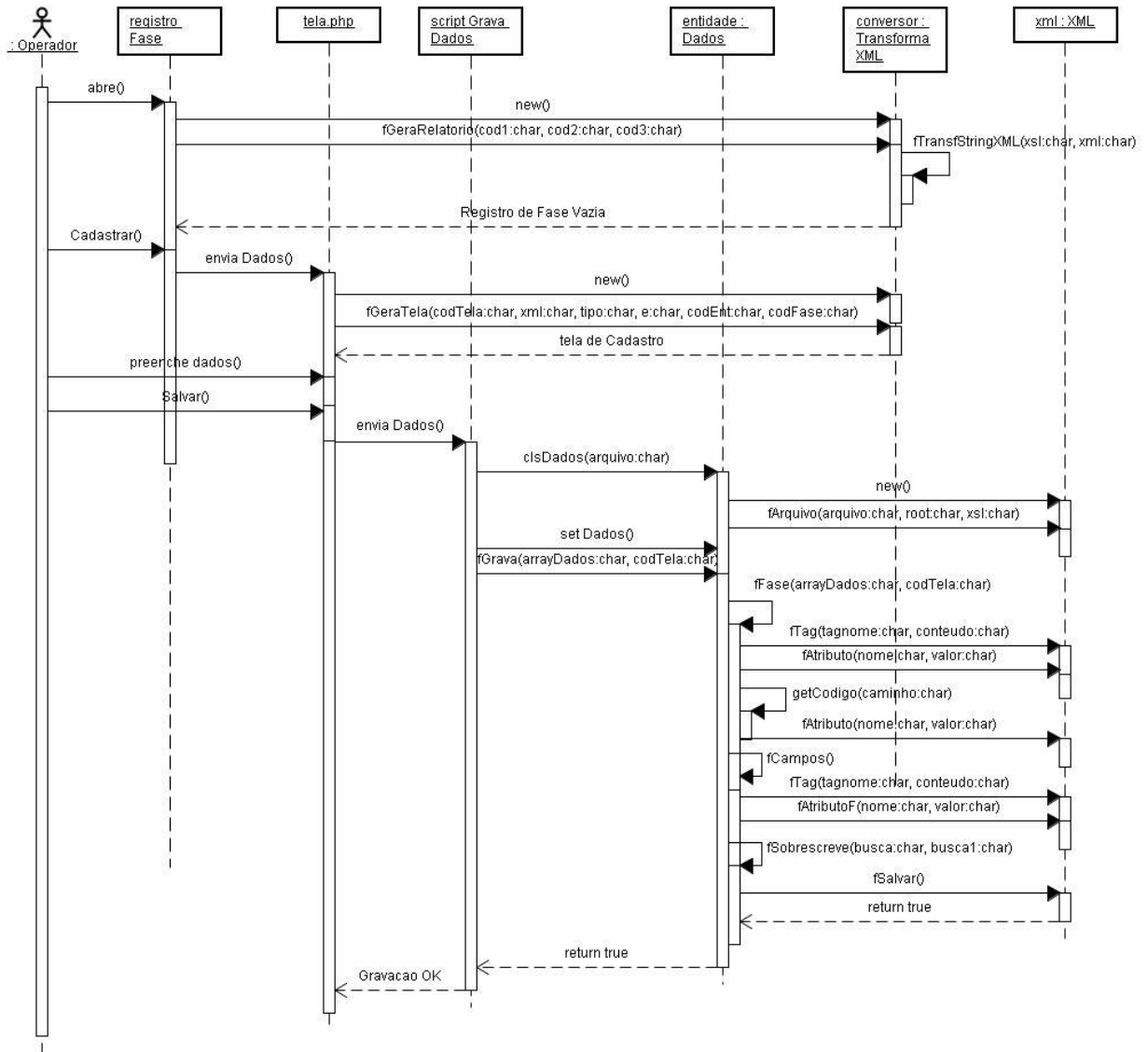


Figura 72 – Seqüência Alterar Registro de Fase

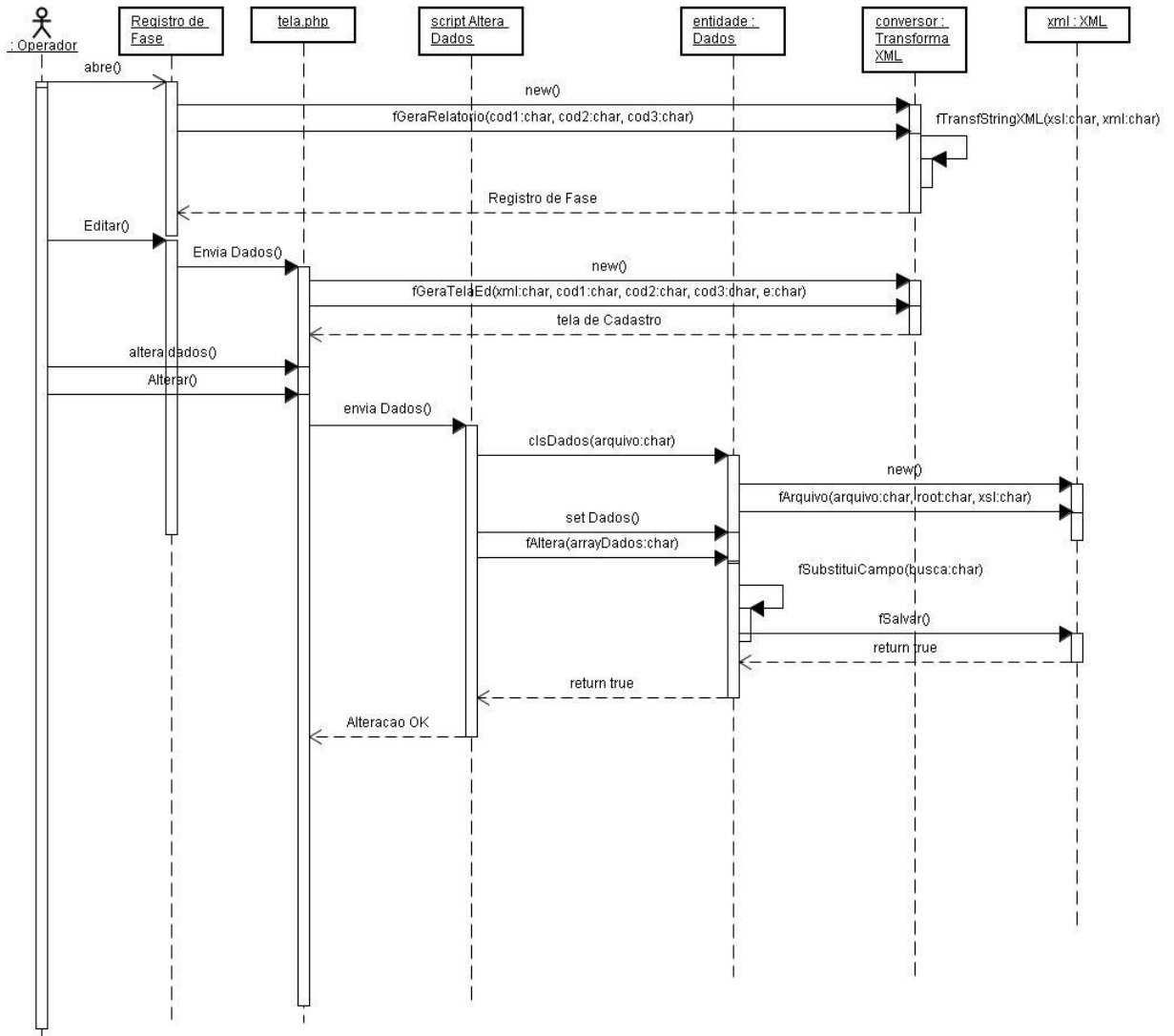


Figura 73 – Colaboração Incluir Registro de Fase

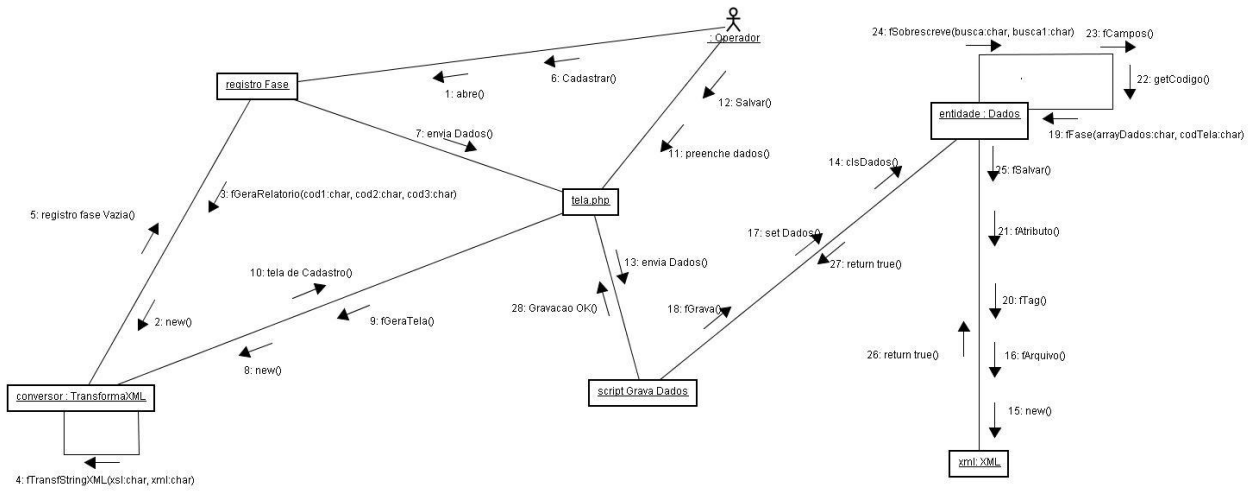


Figura 74 – Colaboração Alterar Registro de Fase

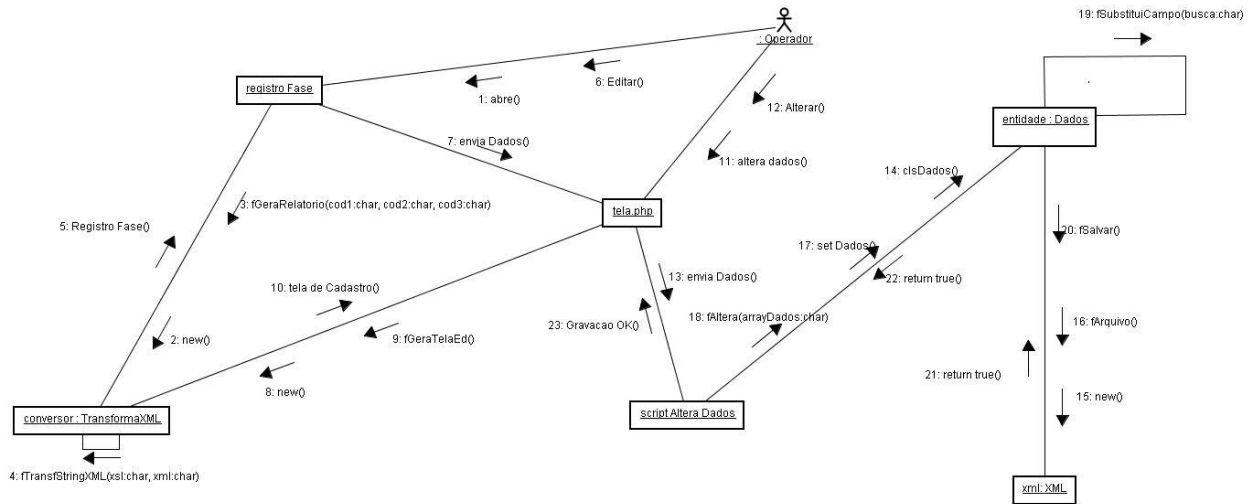


Figura 75 –Registro de Origem

ORIGEM - Produtos Agrícolas	Milho nº: 4
Não há dados cadastrados.	
CADASTRAR	

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 76 – Incluir Registro de Origem

CADASTRO DE ORIGEM - Origem		Milho nº: 4
Propriedade:	<input type="text"/>	
Cooperado:	<input type="text"/>	
Número de matricula:	<input type="text"/>	
Quantidade de semente:	<input type="text"/>	
Cultivar/Varietade:	<input type="text"/>	
Número do lote da semente:	<input type="text"/>	
Fertilizante:	<input type="text"/>	
Herbicida:	<input type="text"/>	
Inseticida:	<input type="checkbox"/> inseticida A <input type="checkbox"/> Inseticida B <input type="checkbox"/> Inseticida C	
Talhão:	<input type="text"/>	
Area Plantada:	<input type="text"/>	
<input type="button" value="Salvar"/>		

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 77 – Seqüência Incluir Registro de Origem

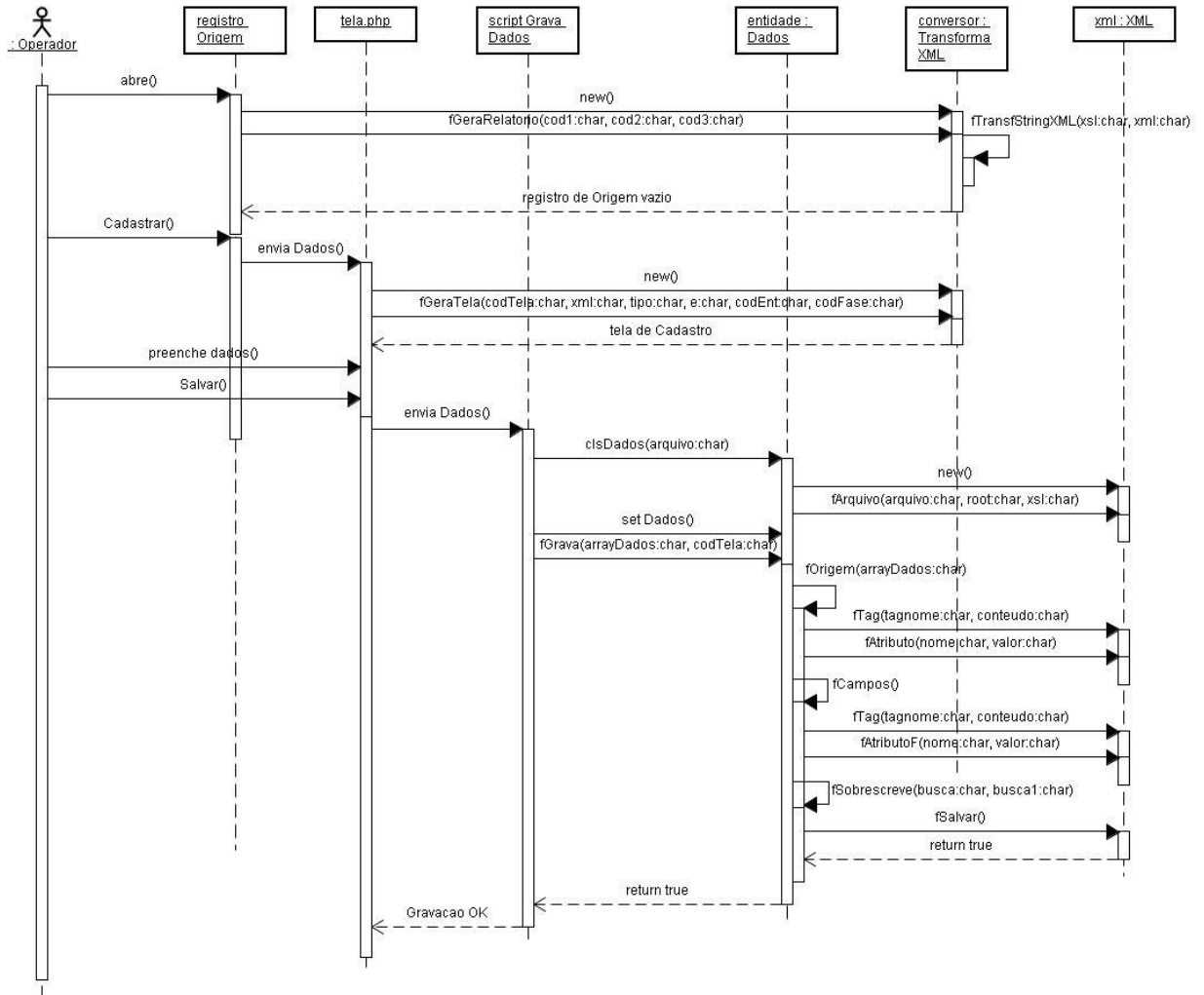


Figura 78 – Seqüência Alterar Registro de Origem

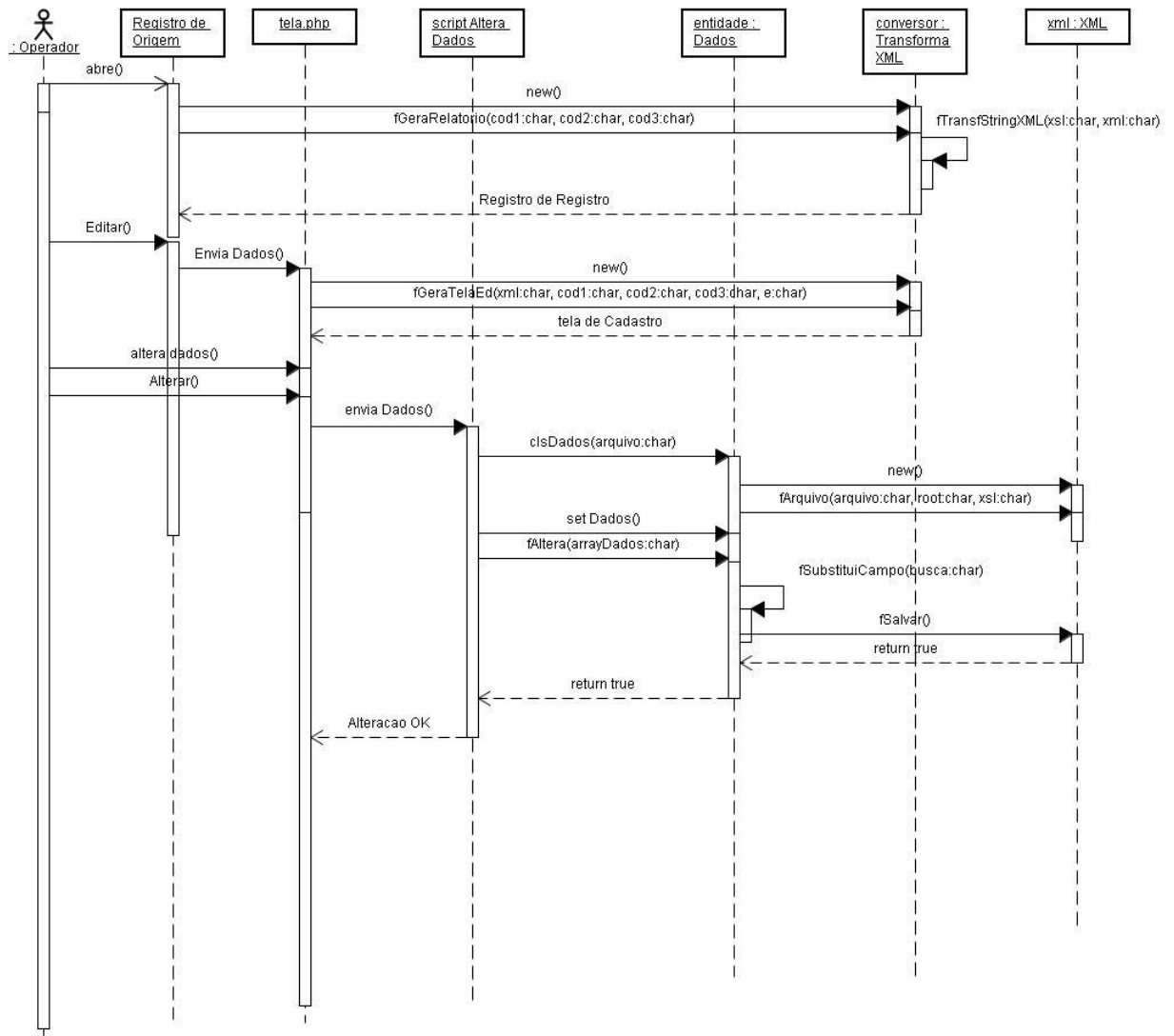


Figura 79 – Colaboração Incluir Registro de Origem

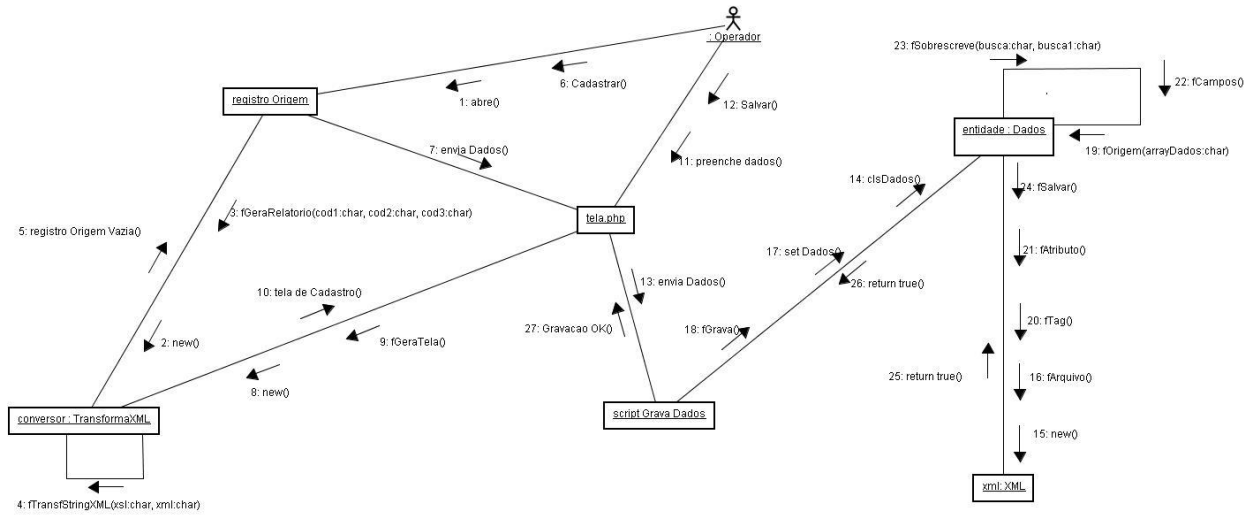


Figura 80 – Colaboração Alterar Registro de Origem

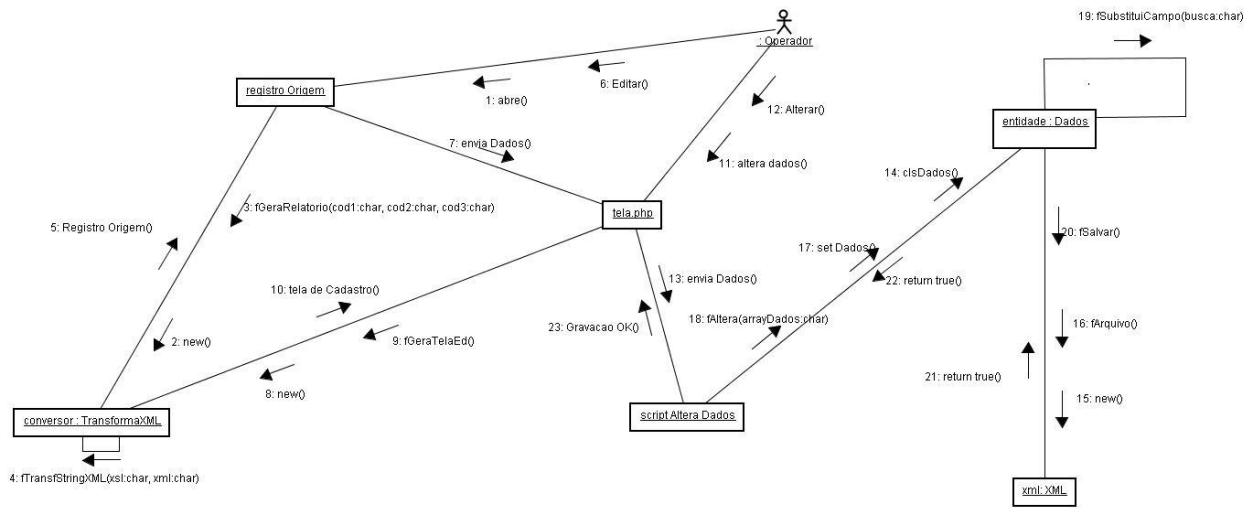


Figura 81 – Registro de Processo

Objeto da rastreabilidade: Milho 6

6
IR

CADASTRO DE PROCESSO - Termometria		Milho nº: 6
Data:	<input type="text"/>	
Temperatura média:	<input type="text"/>	
Temperatura ambiente:	<input type="text"/>	
<input type="button" value="Salvar"/>		

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 82 – Incluir Registro de Processo

Objeto da rastreabilidade: Milho 6

CADASTRO DE PROCESSO - Termometria		Milho nº: 6
Data:	<input type="text"/>	
Temperatura média:	<input type="text"/>	
Temperatura ambiente:	<input type="text"/>	
<input type="button" value="Salvar"/>		

* Tela meramente ilustrativa. A estrutura desta tela deve variar de acordo com que for definido pelo administrador.

Figura 83 – Seqüência Incluir Registro de Processo

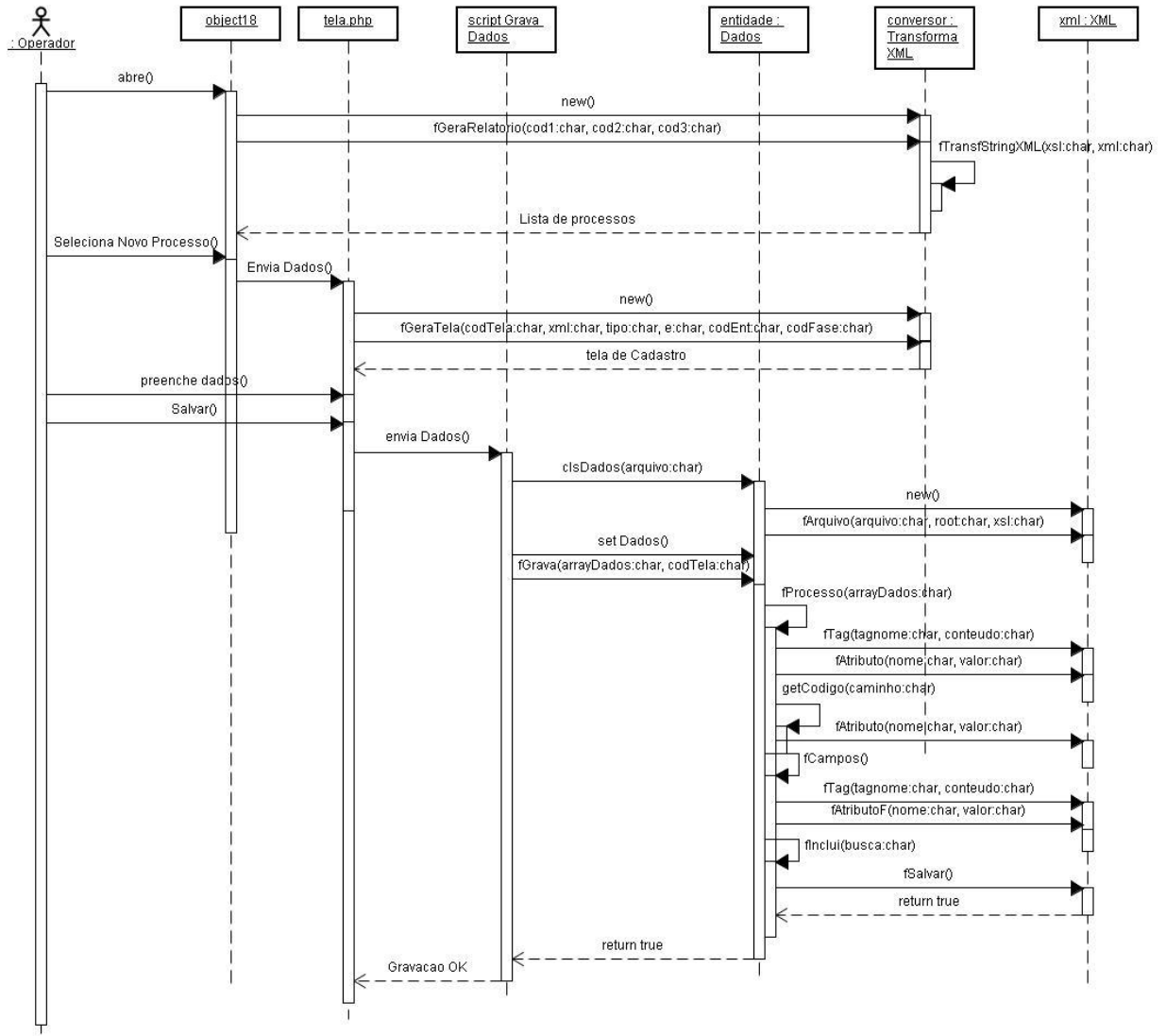


Figura 84 – Seqüência Alterar Registro de Processo

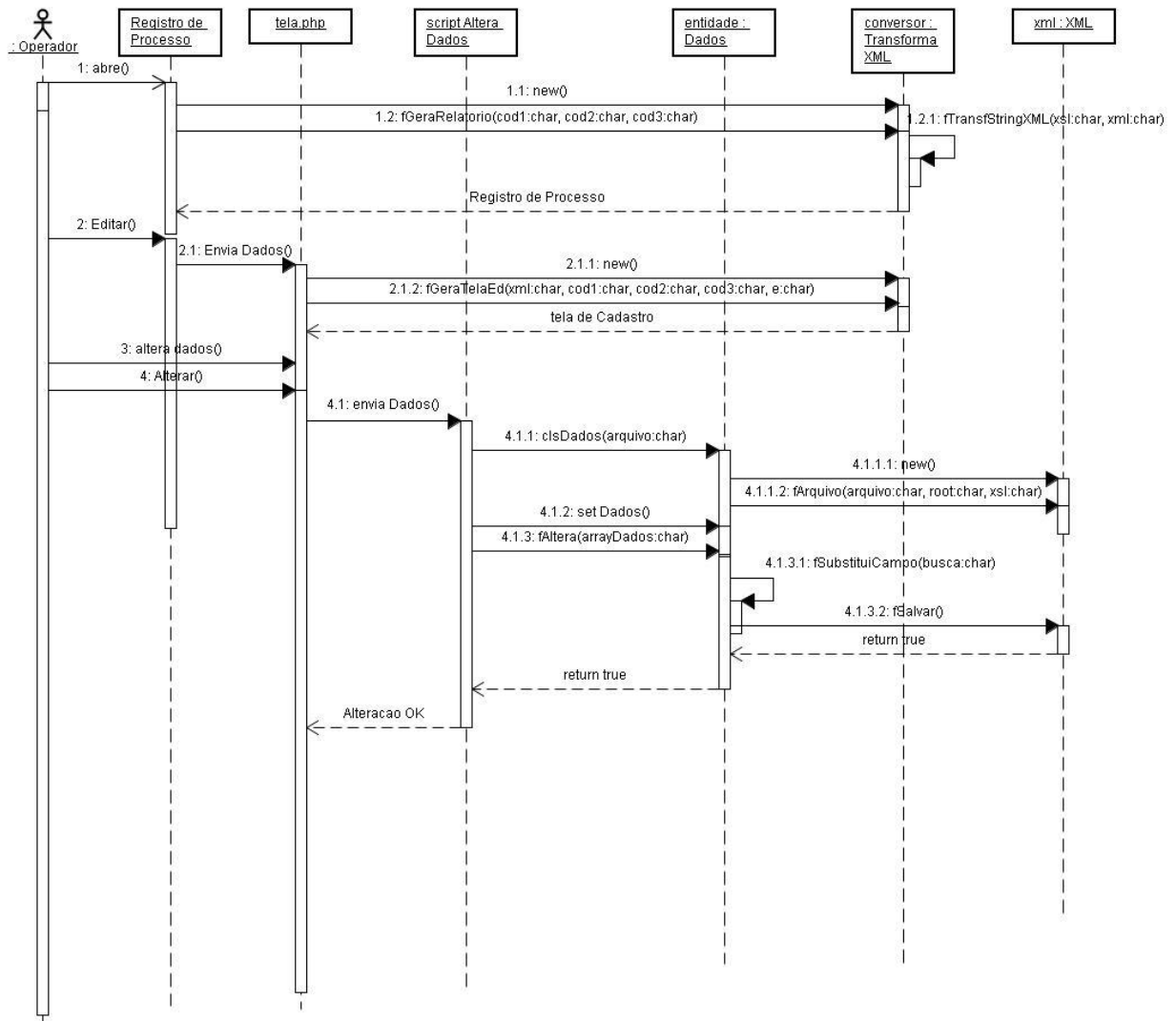


Figura 85 – Seqüência Excluir Registro de Processo

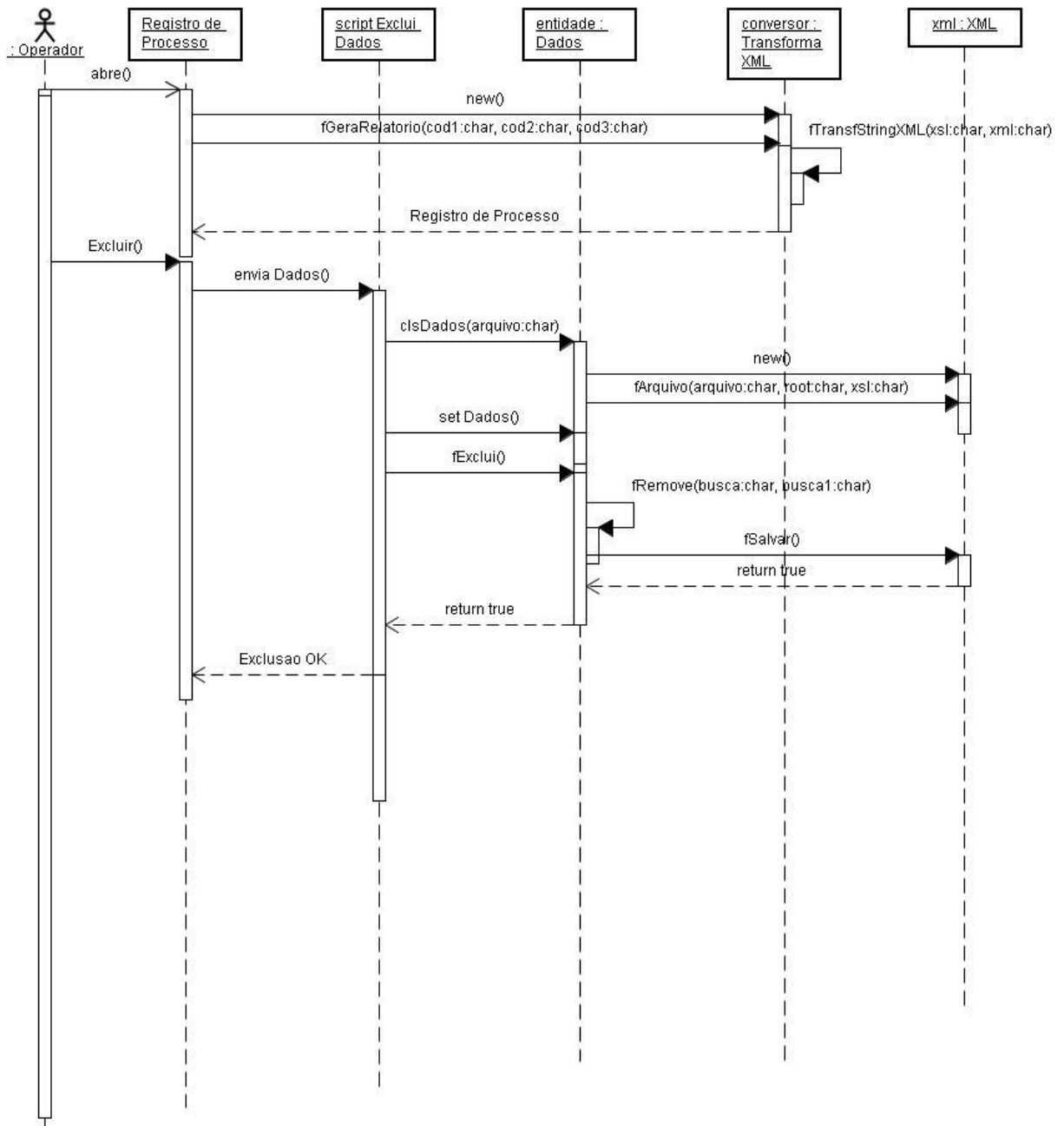


Figura 86 – Colaboração Incluir Registro de Processo

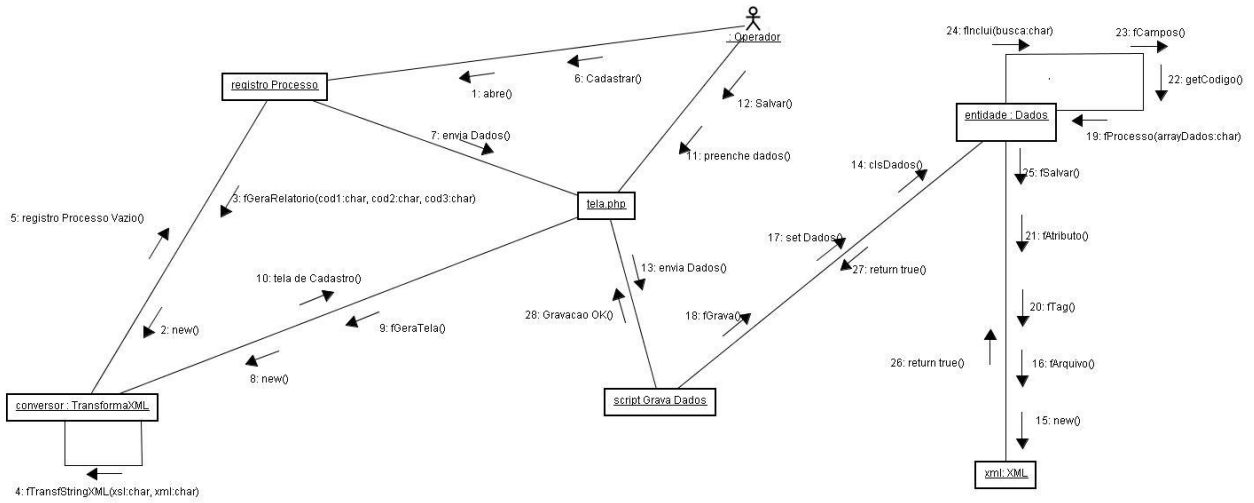


Figura 87 – Colaboração Alterar Registro de Processo

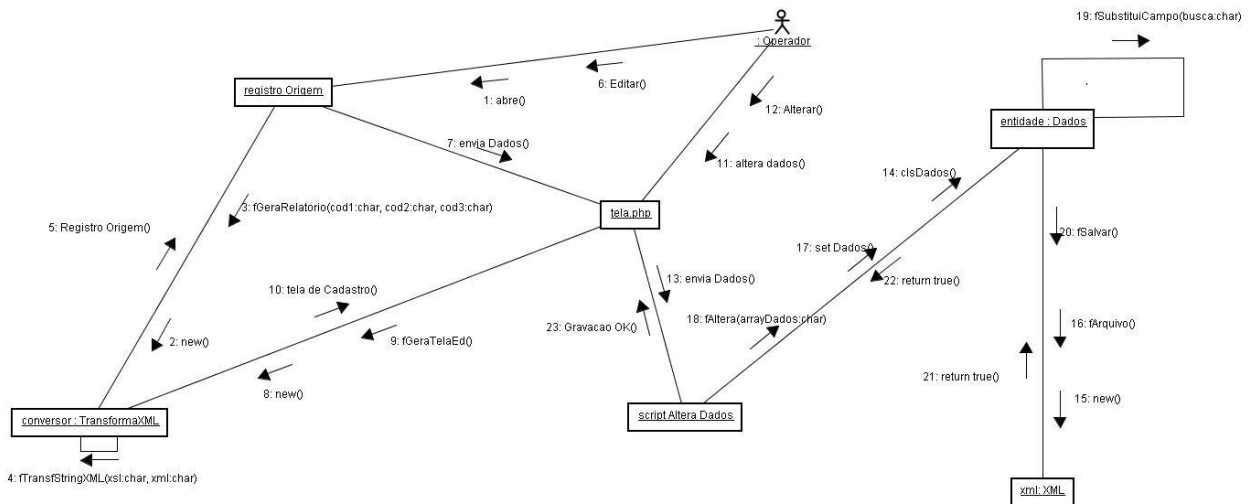
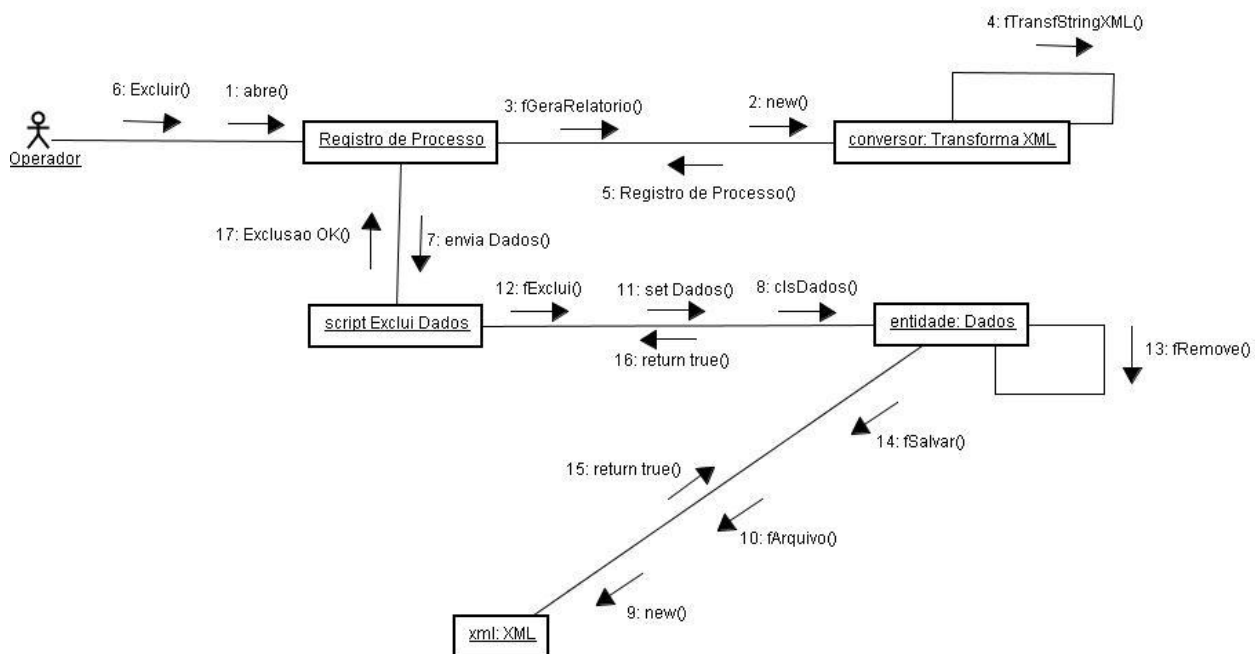


Figura 88 – Colaboração Excluir Registro de Processo



APÊNDICE 8 - CÓDIGO FONTE

TELAS E SCRIPTS

Nome do arquivo: Index.php

Descrição: é a tela de login do sistema, verifica se o login e a senha informados estão corretos, em caso positivo inicia a sessão com os dados do usuário.

```
<?php
session_start();
include_once('classes\clsXML.php');
$titulo = "RASTREABILIDADE";
$letra = "font-family: Tahoma; font-size: 10pt; color: #339966;";
$form = "font-family: Tahoma; font-size: 10pt; color: #A2EE88; border: 1px solid #339966; background:
#339966;";

if (!empty($_POST['login']) && !empty($_POST['senha'])) {
// verificacao dos dados enviados
    $login = $_POST["login"];
    $senha = $_POST["senha"];

    $xml = simplexml_load_file('xml/XMLEstrutura/usuarios.xml');
    foreach($xml as $item)
    {
        if ($item->login==$login && $item->senha==$senha)
        {
            $lognome = $item->nome;
            $tipo = $item->tipo;
            $codusuario = $item['CODIGO'];
            $aux=1;
        } //fim if
    } // fim foreach

if ($aux==1)
{
    $_SESSION['tipo']="$tipo";
    $_SESSION['lognome']="$lognome";
    $_SESSION['login']="$login";
    $_SESSION['senha']="$senha";
    $_SESSION['codigo']="$codusuario";

    if($tipo=="Administrador"){?>
<script language="javascript">
        document.location = "telas/principal.php";
</script>
<? }
    else{
    ?>
```

```

        <script language="javascript">
            document.location = "telas/principal_user.php";
        </script>
        <? }
        exit;
    }
    else {
        echo "<center>\n<br><br><br><br><br>\n<font style='$letra'>\nDados
errados!!\n<br><br><br>\n<a href='index.php'>Voltar</a>\n</font>\n</center>";
        exit;
    }
} else {

?>
<html>
<head>
<title><?php echo"$titulo"; ?></title>
</head>
<body bgcolor="#FFFFFF">
<form method="post" action="index.php">
    <p>&nbsp;</p>
    <table width="67%" height="300" border="0" align="center" cellpadding="0" cellspacing="0">
<tr bgcolor="#FFFFFF">
    <td width="131" height="131"><p align="center"></p>
    <p align="center"></p></td>
    <TD width="369"><p><strong><font style="<? echo "$letra"; ?>"><font color="#339966"
face="Arial, Helvetica, sans-serif">Seja Bem vindo ao Sistema de Rastreabilidade!<br>
    </font></font></strong></p>
    <p><strong><font style="<? echo "$letra"; ?>"><font color="#339966" face="Arial, Helvetica,
sans-serif"> Acesso restrito</font></font></strong><font color="#339966" face="Arial, Helvetica, sans-
serif">!</font><font face="Arial, Helvetica, sans-serif"></font></p></TD>
</tr>
<tr bgcolor="#FFFFFF">
    <td colspan="2"><div align="center">
        <p>&nbsp;</p>
        </div>

        <div align="center"><strong><font color="#339966" face="Arial, Helvetica, sans-serif" style="<? echo
"$letra"; ?>">Nome</font></strong><font style="<? echo "$letra"; ?>"><br>
        <input name="login" type="Text" id="login" style="<? echo "$form"; ?>">
        <br>
        <font color="#339966" face="Arial, Helvetica, sans-serif"><strong>Senha</strong></font><br>
        <input type="password" name="senha" style="<? echo "$form"; ?>">
        <br>
        <br>
        <input type="Submit" name="submit" value="Entrar" style="<? echo "$form"; ?>">
        </font></div>
    <?php
} // end if
?> <p>&nbsp;</p></td>
</tr>
</table>
</form>

```

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

Nome do arquivo: login.inc

Descrição: este arquivo é chamado toda vez que uma tela do sistema é aberta, se existirem dados do usuário na sessão significa que o sistema está logado, caso contrário redireciona o sistema para a tela de login.

```
<?
session_start();
if (empty($_SESSION['login']) || empty($_SESSION['senha']))
{?>
<script language="javascript">
  document.location = "../index.php";
</script>
<?
}
?>
```

Nome do arquivo: Campos.php

Descrição: código da tela de campos, usada para incluir, alterar e excluir novos campos.

```
<?php
  include("login.inc");
  if($_SESSION['tipo']!="Administrador"){?>
  <script language="javascript">
    document.location = "principal_user.php";
  </script>
<? }?>

<script language="JavaScript" type="text/javascript">

  window.defaultStatus=".: Sistema de Rastreabilidade";
  document.onmousedown=verfonte;

  function verfonte()
  {
    if (event.button==2)
      {
        window.alert('Todos os direitos reservados - Ana e Eduardo')
      }
  } //fim ver fonte

  function Selecciona(hdn,cbx)
  {
    for (var i=0; i<cbx.length; i++)
```

```

{
    if (cbx.options[i].value == hdn.value)
    {
        cbx.options[i].selected = true
    }
} //fim for
} //fim fuction

function fValidaCampos()
{
    var f1 = document.CadCampos;
    if (f1.edtNome.value.length == 0)
    {
        window.alert("O Campo 'Nome' precisa ser preenchido");
        f1.edtNome.focus();
        return false;
    }
}

function mensagem(){
    var a = document.getElementById("lboxTipoHtml");
    if(a.options[a.selectedIndex].value=="select" || a.options[a.selectedIndex].value=="checkbox"){
        alert("As opções devem ser preenchidas no campo 'Valor default' separados por ','");
    }
}
}

</script>

    <td valign="top">
<?
include_once('../classes/clsCampos.php');
$campo = $_REQUEST["campo"];

if ($campo > "0" and $campo < "1000000000")
{
    $get = new clsCampos($campo,'../xml/bancocampo.xml');
    $opcao = 2;
}
else
{
    $get = new clsCampos("", '../xml/bancocampo.xml');
    $opcao = 1;
}
?>

<form name="CadCampos" method="post" action="scripts/scriptCampos.php?opcao=<?echo
$opcao;?>&campo=<?echo $campo;?>" onSubmit="return fValidaCampos()">

    <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0">
    <tr bgcolor="#339933">
    <th height="22" colspan="2" bgcolor="#339933"><div align="left"><font color="#FFFFFF"
face="Verdana, Arial, Helvetica, sans-serif"><strong><font face="Verdana, Arial, Helvetica, sans-
serif">Cadastro

```


Nome do arquivo: lista_campos.xsl

Descrição: arquivo que transforma o arquivo bancocampo.xml em uma tabela na tela
Cadastrar Campos.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>
<head>
<title>Lista de campos</title>
</head>

<style type="text/css">
a:link {
    color: #006600;
    text-decoration: none;
}
a:visited {
    text-decoration: none;
    color: #006666;
}
a:hover {
    text-decoration: none;
    color: #669933;
}
a:active {
    text-decoration: none;
    color: #669933;
}
</style>

<body>
<br></br>

    <table border="0" align="center" width="100%">
        <tr><th colspan="10" bgcolor="#339933"><font color="#FFFFFF" face="Verdana, Arial, Helvetica,
sans-serif">Lista de campos </font></th></tr>
        <tr bgcolor="#339933">
            <th width="" bgcolor="#339933"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif"></font></th>
            <th width="25%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif">Nome</font></th>
            <th width=""><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif">Tipo de
html</font></th>
            <th width="25%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif">Valor
Default</font></th>
            <th width=""><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"></font></th>
        </tr>
```

```

<xsl:for-each select="banco_campos/campo">
  <xsl:sort select="nome"/>
  <xsl:if test="@ATIVO='S'">
    <tr>
      <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:number value="position()" format="1. "/></b></font></td>
      <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><a ><xsl:attribute name="href">Campos.php?campo=<xsl:value-of select="@CODIGO"/></xsl:attribute><xsl:value-of select="nome" /></a></b></font></td>
      <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:value-of select="tphtml"/></b></font></td>
      <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b>
        <xsl:for-each select="valores/opcao"><xsl:value-of select="."/> </xsl:for-each></b></font></td>
      <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><a><xsl:attribute name="href">scripts/scptExcluiCampos.php?campo=<xsl:value-of select="@CODIGO"/></xsl:attribute>Apagar</a></b></font></td>
    </tr>
  </xsl:if>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Nome do arquivo: scriptCampos.php

Descrição: este script é chamado quando o usuário manda incluir ou alterar um cadastro de campo, de acordo com a opção enviada, chama as funções incluir() ou alterar() da classe campos.

```

<?
include_once('../classes/clsCampos.php');

$opcao = $_REQUEST['opcao'];
$codigo = $_REQUEST['campo'];
$campo = new clsCampos(null, '../xml/bancocampo.xml');

$campo->setNome($_POST['edtNome']);
$campo->setTphtml($_POST['lboxTipoHtml']);
$campo->setDefault($_POST['edtValorDefault']);

if ($opcao == 1){
    $res = $campo->incluir();
}
elseif ($opcao == 2){
    $res = $campo->alterar($codigo);
}
if (res != "0"){?>
<script language='JavaScript'>

```

```

        window.alert("Operação efetuada com sucesso!");
        window.parent.location = '../Campos.php?campo=N';
    </script>
    <?
    }
    ?>

```

Nome do arquivo: sctExcluiCampos.php

Descrição: script chamado quando o usuário clica no link excluir, da tela Cadastrar Campo, chama a função excluir() da classe campos.

```

<?
include_once('../classes/clsCampos.php');
//Código da Página.
$codigo = $_REQUEST['campo'];
$campo = new clsCampos("", '../xml/bancocampo.xml');
$res = $campo->excluir($codigo);

if ($res == true){?>
<script language='JavaScript'>
    window.alert("Exclusão efetuada com sucesso!");
    window.parent.location = '../Campos.php';
</script>
<?
}
?>

```

Nome do arquivo: Usuarios.php

Descrição: código da tela onde são feitas a inclusão, a alteração e a exclusão de usuários.

```

<?php
    include("login.inc");
    if($_SESSION['tipo']!="Administrador"){?>
    <script language="javascript">
        document.location = "principal_user.php";
    </script>
    <? }?>
    <script language="JavaScript" type="text/javascript">

        window.defaultStatus=".: Sistema de Rastreabilidade";
        document.onmousedown=verfonte;

        function verfonte()
        {
            if (event.button==2)

```

```

        {
            window.alert('Todos os direitos reservados - Ana e Eduardo')
        }
    } //fim ver fonte

function Selecciona(hdn,cbx)
{
for (var i=0; i<cbx.length; i++)
{
    if (cbx.options[i].value == hdn.value)
    {
        cbx.options[i].selected = true
    }
} //fim for
} //fim fuction

function fValidaCampos()
{
    var f1 = document.CadUsuarios;
    if (f1.edtNome.value.length == 0)
    {
        window.alert("O Campo 'Nome' precisa ser preenchido");
        f1.edtNome.focus();
        return false;
    }
}

</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</td>
    <td valign="top">
        <?
            include_once('../classes/clsUsuario.php');
            $codigo = $_REQUEST["cod"];
            if ($codigo != "")
            {
                $usuario = new clsUsuarios($codigo,'../xml/usuarios.xml');
                $opcao = 2;
            }
            else
            {
                $usuario = new clsUsuarios("0",'../xml/usuarios.xml');
                $opcao = 1;
            }
        ?>
<form name="CadUsuarios" method="post" action="scripts/scriptUsuario.php?opcao=<?echo
$opcao;?>&codigo=<?echo $codigo;?>" onSubmit="return fValidaCampos()">
    <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0">
        <tr bgcolor="#339933">
            <th height="22" colspan="2"><div align="left"><font color="#FFFFFF" face="Verdana, Arial, Helvetica,
sans-serif"><strong><font face="Verdana, Arial, Helvetica, sans-serif">Cadastro
de Usu&aacute;rios</font></strong> </font> </div></th><th width="10%"><div><font
size="2"><a href="principal.php" target="_top"><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"></font></a></font>&nbsp;&nbsp;&nbsp;&nbsp;</div></th>

```


Nome do arquivo: lista_usuarios.xsl

Descrição: arquivo XSL que transforma o arquivo usuario.xml em uma tabela na tela
Cadastrar Usuário.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>
<head>
<title>Lista de campos</title>
</head>

<style type="text/css">
a:link {
    color: #006600;
    text-decoration: none;
}
a:visited {
    text-decoration: none;
    color: #006666;
}
a:hover {
    text-decoration: none;
    color: #669933;
}
a:active {
    text-decoration: none;
    color: #669933;
}
</style>

<body>
<br></br>

    <table border="0" align="center" width="100%">

        <tr><th colspan="10" bgcolor="#339933"><font color="#FFFFFF" face="Verdana, Arial, Helvetica,
sans-serif">Lista de usuários </font></th></tr>
        <tr bgcolor="#339933">
            <th width="5%" bgcolor="#339933"><font color="#FFFFFF" face="Verdana, Arial, Helvetica,
sans-serif"></font></th>
            <th width="30%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif">Nome</font></th>
            <th width="25%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif">Login</font></th>
            <th width="25%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif">Tipo de
usuário</font></th>
            <th width="15%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif"></font></th>
```

```

</tr>

    <xsl:for-each select="usuarios/usuario">
    <xsl:sort select="nome"/>
    <xsl:if test="@ATIVO='S'">
    <tr>
        <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:number value="position()" format="1. "/></b></font></td>
        <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><a ><xsl:attribute name="href">Usuarios.php?cod=<xsl:value-of select="@CODIGO"/></xsl:attribute><xsl:value-of select="nome" /></a></b></font></td>
        <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:value-of select="login"/></b></font></td>
        <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:value-of select="tipo"/></b></font></td>
        <td bgcolor="#66CC66"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><a><xsl:attribute name="href">scripts/scriptExcluiUsuario.php?cod=<xsl:value-of select="@CODIGO"/></xsl:attribute>Apagar</a></b></font></td>
    </tr>
    </xsl:if>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Nome do arquivo: scriptUsuario.php

Descrição: arquivo chamado quando o Administrador manda incluir ou alterar um cadastro de usuário. De acordo com a opção enviada, chama as funções Incluir() ou alterar() da classe usuário.

```

<?
include_once('../classes/clsUsuario.php');
//Código da Página.
$opcao = $_REQUEST['opcao'];
$codigo = $_REQUEST['codigo'];
$Usuario = new clsUsuarios("0",'../xml/usuarios.xml');
$Usuario->setNome($_POST['edtNome']);
$Usuario->setTipo($_POST['lboxTipo']);
$Usuario->setLogin($_POST['edtLogin']);
$Usuario->setSenha($_POST['edtSenha']);
if ($opcao == 1){
    $res = $Usuario->incluir();
}
elseif ($opcao == 2){
    $res = $Usuario->alterar($codigo);
}

if ($res == true){?>

```

```

<script language='JavaScript'>
    window.alert("Operação efetuada com sucesso!");
    window.parent.location = '../Usuarios.php';
</script>
<?
}
?>

```

Nome do arquivo: scriptExcluiUsuario.php

Descrição: script chamado quando o Administrador clica no link Apagar, da tela Cadastrar Usuário.

```

<?
include_once('../classes/clsUsuario.php');
//Código da Página.
$codigo = $_REQUEST['cod'];
$usuario = new clsUsuarios("0", '../xml/usuarios.xml');
$res = $usuario->excluir($codigo);
if ($res == true){?>
<script language='JavaScript'>
    window.alert("Exclusão efetuada com sucesso!");
    window.parent.location = '../Usuarios.php';
</script>
<?
}
?>

```

Nome do arquivo: altera_senha.php

Descrição: código que monta a tela de alteração de senha e, quando usuário manda salvar, recebe os dados e chama a função alterar da classe usuario.

```

<script language="JavaScript" type="text/javascript">
    window.defaultStatus=".: Sistema de Rastreabilidade";
    document.onmousedown=verfonte;
    function verfonte()
    {
        if (event.button==2)
        {
            window.alert('Todos os direitos reservados - Ana e Eduardo')
        }
    } //fim ver fonte

    function fValidaSenha()
    {
        var codigo = <?echo $_SESSION['senha'];?>;
        if(codigo == document.CadUsuarios.edtSenha.value){

```

```

        if(document.CadUsuarios.edtSenha2.value == document.CadUsuarios.edtSenha3.value){
            return true;
        }
        else{
            alert("Confirmação de senha não confere!");
            return false;
        }
    }
    else{
        alert("A senha atual não confere!");
        return false;
    }
}
</script>
<?
    include_once('../classes/clsUsuario.php');
    $usuario = new clsUsuarios($_SESSION['codigo'],'./xml/usuarios.xml');
    if($_REQUEST['grava']=="S"){
        $usuario->setSenha($_POST['edtSenha2']);
        if ($usuario->alterar($_SESSION['codigo'])){
            ?>
            <script language='JavaScript'>
                window.alert("OperaÃ§Ão efetuada com sucesso!");
            </script>
            <?
            }
        }
    ?>
    <form name="CadUsuarios" method="post" action="altera_senha.php?grava=S" onSubmit="return
fValidaSenha()">
    <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0">
    <tr bgcolor="#339933">
    <th height="22" colspan="2"><div align="left"><font color="#FFFFFF" face="Verdana, Arial,
Helvetica, sans-serif"><strong><font face="Verdana, Arial, Helvetica, sans-serif">Altera&cedil;&atilde;o
de senha </font></strong></font> </div></th>
    <th width="10%"><div>&nbsp;&nbsp;&nbsp;</div></th>
    </tr>
    <tr>
    <td width="22%"><strong><font size="2" face="Verdana, Arial, Helvetica, sans-
serif">Nome:</font></strong></td>
    <td width="68%"><? echo $usuario->getNome();?> </td>
    </tr>
    <tr>
    <td><strong><font size="2" face="Verdana, Arial, Helvetica, sans-
serif">Login:</font></strong></td>
    <td><? echo $usuario->getLogin();?></td>
    </tr>
    <tr>
    <td height="28"><strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">Senha
atual:</font></strong></td>
    <td><input name="edtSenha" type="password" id="edtSenha" value=""></td>
    </tr>
    <tr>
    <td><strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">Nova
Senha:</font></strong></td>

```



```

</tr>
  <tr><td colspan="3" >
    <a><xsl:attribute name="href">../telas/constroi_Entidade.php</xsl:attribute><font
color="#339999" face="Verdana, Arial, Helvetica, sans-
serif"><strong><u>Nova</u></strong></font></a><br /><br /></td></tr>

    <xsl:for-each select="banco_telas/ENTIDADE">
    <xsl:if test="@ATIVO='S'">
    <tr bgcolor="#66CC66">
      <td width="80%"><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-
serif"><b><xsl:value-of select="@nome"/></b></font></td>
      <td align="center">
        <a><xsl:attribute name="href">../telas/constroi_Entidade.php?tela=<xsl:value-of
select="@CODIGO"/></xsl:attribute>
        <font color="006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Editar</b></font>
        </a></td>
      <td align="center">
        <a><xsl:attribute name="href">../telas/scripts/ExcluiEntidade.php?tela=<xsl:value-of
select="@CODIGO"/></xsl:attribute>
        <font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Apagar</b></font>
        </a></td>
    </tr>
    </xsl:if>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Nome do arquivo: constroi_Entidade.php

Descrição: código que monta a tela de cadastro e edição de acervo de Entidade. Quando o usuário clica em salvar, recebe os dados e, de acordo com a opção enviada, chama a função fSalvaTela() ou editar() da classe Acervo (tela).

```

<?php
  include("login.inc");
  include_once('../classes/clsCampos.php');
  include_once('../classes/clsTela.php');
  if($_SESSION['tipo']!="Administrador"){?>
  <script language="javascript">
    document.location = "principal_user.php";
  </script>
<? }
  $opcao="S";
  $tela = new clsTela();
  if ($_REQUEST["Salva"]=="S"){
    $tela->SetXML("../xml/bancoentidades.xml");
    $tela->SetTipo($_POST["tpTela"]);
    $tela->SetCampos($_POST["hdnDados"]);
  }

```

```

        $tela->SetMetadado($_POST["hdnFases"]);
        $tela->SetOrigem($_POST["cbxOrigem"]);
        $tela->SetNome($_POST["edtNome"]);
        if ($tela->fSalvaTela()){
            ?>
            <script language="JavaScript">
                alert("Tela gravada com sucesso!");
                document.location ="bancoentidade.php";
            </script>
            <?
        }
    }elseif($_REQUEST["Salva"]=="E"){
        $tela->SetXML("../xml/bancoentidades.xml");
        $tela->SetTipo($_POST["tpTela"]);
        $tela->SetCampos($_POST["hdnDados"]);
        $tela->SetMetadado($_POST["hdnFases"]);
        $tela->SetOrigem($_POST["cbxOrigem"]);
        $tela->SetNome($_POST["edtNome"]);
        if ($tela->editar($_REQUEST["tela"])){
            ?>
            <script language="JavaScript">
                alert("Tela alterada com sucesso!");
                document.location ="bancoentidade.php";
            </script>
            <?
        }
    }elseif($_REQUEST["tela"]!=""){
        $tela->SetXML("../xml/bancoentidades.xml");
        $tela->setTela($_REQUEST["tela"]);
        $opcao="E";
    }
}
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Sistema rastreabilidade</title>
<script language="JavaScript" type="text/javascript">

    window.defaultStatus=".: Sistema de Rastreabilidade";
    document.onmousedown=verfonte;

    function verfonte()
    {
        if (event.button==2)
        {
            window.alert('Todos os direitos reservados - Ana e Eduardo')
        }
    }
} //fim ver fonte

vetCampos = new Array(); //variável que será usada para gravar os dados no banco
vetFases = new Array(); //variável que será usada para gravar as fases no banco

<?
    $codCampos = explode(",",$tela->getCampos());

```

```

    $cont = count($codCampos);
    for ($x=0;$x<$cont-1;$x++){
        echo "add = vetCampos.splice($x,0,$codCampos[$x]); \n";
    }
    if($tela->GetMetadado()!=""){
        $codMetadado = explode(",",$tela->GetMetadado());
        $cont = count($codMetadado);
        for ($x=0;$x<$cont;$x++){
            echo "add = vetFases.splice($x,0,$codMetadado[$x]); \n";
        }
    }
?>

```

```

cont=-1; //cont recebe -1 para ser usado abaixo
function adCampo(index) //index serve para indicar a posição do campo no vetor que deve ser o
mesmo do select
{
    var i=document.getElementById('ordem').value // captura o valor do campo ordem para inserir o
registro na ordem indicada

    var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
    var a=document.getElementById("cbxCampos") //captura o select cbxlista p/ uma variável a
        var dados= a.options[a.selectedIndex].value.split("|") //transforma um campo de aTPhTml em
um novo array chamado dados
    add = vetCampos.splice(i-1,0,dados[0])
    var y=x.insertCell(0) //insere as quatro células da tabela
    var z=x.insertCell(1)
    var v=x.insertCell(2)
    y.innerHTML=(a.options[a.selectedIndex].text) //altera o valor do html colocando o valor do texto
do item selecionado no select
    if (dados[2] == 'checkbox'){
        var opcoes= dados[3].split(";")
        var cont=0
        for (cont;cont<opcoes.length;cont++){
            z.innerHTML=z.innerHTML+"<input type=checkbox name="+opcoes[cont]+
value=checkbox disabled>"+opcoes[cont]+<p>"
        }
    }
    if (dados[2] == 'textarea'){
        z.innerHTML="<textarea name="+dados[1]+ disabled>"+dados[3]+</textarea>"
    }
    if (dados[2] == 'select'){

        var opcoes= dados[3].split(";")
        var cont=0
        var vselect
            vselect="<select name=select1>"
        for (cont;cont<opcoes.length;cont++){
            vselect=vselect+"<option
value="+opcoes[cont]+>"+opcoes[cont]+</option>"
        }
        z.innerHTML=vselect+"</select>"
    }
}

```

```

        if (dados[2] == 'text') {
            z.innerHTML="<input type='text' name="+dados[1]+" disabled value="" +dados[3]+">"
        }
        v.innerHTML="<a href=# onclick='deleteRow(this.parentNode.parentNode.rowIndex)'\>Excluir</a>"
        document.getElementById('ordem').value++
    }
    function edCampo() //index serve para indicar a posi o do campo no vetor que deve ser o
    mesmo do select
    {
        var cbx = document.getElementById("cbxCampos");
        var a;
        for(a=0;a<=vetCampos.length-1;a++){ //for 01
            var b=0;
            for(b; b<cbx.length;b++){
                var teste= cbx[b].value.split("|"); //transforma um campo de aTphTml em um novo
                array chamado teste
                if(teste[0]==vetCampos[a]){
                    dados = cbx[b].value.split("|");
                }
            }
            var i= document.getElementById('ordem').value; // captura o valor do campo ordem
            para inserir o registro na ordem indicada
            var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
            var y=x.insertCell(0) //insere as quatro c lulas da tabela
            var z=x.insertCell(1)
            var v=x.insertCell(2)
            y.innerHTML=(dados[1]) //altera o valor do html colocando o valor do texto do item
            selecionado no select
            if (dados[2] == 'checkbox'){
                var opcoes= dados[3].split(";")
                var cont=0
                for (cont;cont<opcoes.length;cont++){
                    v.innerHTML=v.innerHTML+"<input type=checkbox name="+opcoes[cont]+"
                    value=checkbox disabled>"+opcoes[cont]+"<p>"
                }
            }
            if (dados[2] == 'textarea'){
                v.innerHTML="<textarea name="+dados[1]+" disabled></textarea>"
            }
            if (dados[2] == 'select'){
                var opcoes= dados[3].split(";")
                var cont=0
                var vselect
                vselect="<select name=select1>"
                for (cont;cont<opcoes.length;cont++){
                    vselect=vselect+"<option value="+opcoes[cont]+">"+opcoes[cont]+"</option>"
                }
                v.innerHTML=vselect+"</select>"
            }
            if (dados[2] == 'text') {
                v.innerHTML="<input type='text' name="+dados[1]+" disabled>"
            }
        }
    }

```

```

        z.innerHTML="<a href=#
onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
        document.getElementById('ordem').value++

    } //fim for 01
}
var i =0;
function adMetadado() //index serve para indicar a posiÃ§Ã£o do campo no vetor que deve ser o
mesmo do select
{
    var x=document.getElementById('tblFases').insertRow(i+1);//insere a linha na tabela campos
    var a=document.getElementById("cbxFases"); //captura o select cbxlista p/ uma variÃ¡vel a
    add = vetFases.splice(i,0,a.value);

    var y=x.insertCell(0); //insere as quatro cÃ©lulas da tabela
    var v=x.insertCell(1);

    y.innerHTML=(a.options[a.selectedIndex].text); //altera o valor do html colocando o valor do texto
do item selecionado no select
    v.innerHTML="<a href=#
onclick='deleteMeta(this.parentNode.parentNode.rowIndex)'>Excluir</a>";
    i++;
}
function edMetadado() //index serve para indicar a posiÃ§Ã£o do campo no vetor que deve ser o
mesmo do select
{
    var c=document.getElementById("cbxFases"); //captura o select cbxlista p/ uma variÃ¡vel a
    var a;
    for(a=0;a<=vetFases.length-1;a++){ //for 01
        for(b=0;b<c.length-1;b++){
            if(vetFases[a]==c.options[b].value){
                var nome=c.options[b].text;
            }
        }
        var x=document.getElementById('tblFases').insertRow(i+1);//insere a linha na tabela campos
        var y=x.insertCell(0); //insere as quatro cÃ©lulas da tabela
        var v=x.insertCell(1)
        y.innerHTML=nome; //altera o valor do html colocando o valor do texto do item selecionado
no select
        v.innerHTML="<a href=#
onclick='deleteMeta(this.parentNode.parentNode.rowIndex)'>Excluir</a>";
        i++;
    }
}

function deleteRow(i)
{
    document.getElementById('campos').deleteRow(i)
    document.getElementById('ordem').value--
    de_fora = vetCampos.splice(i-1,1)
    //alert(vetCampos)
}

function deleteMeta(index)

```

```

{
    document.getElementById('tblFases').deleteRow(index)
    de_fora = vetFases.splice(index-1,1)
    i--;
}

//função para colocar o array numa variável antes de submeter
function enviaarray()
{
    document.getElementById('hDados').value=vetCampos;
    document.getElementById('hFases').value=vetFases;
}

function fValidaCampos()
{
    if(document.cadTela.edtNome.value.length == 0)
    {
        window.alert("O Campo 'Nome' precisa ser preenchido");
        document.cadTela.edtNome.focus();
        return false;
    }
}

function Selecciona(hdn,cbx)
{
    for (var i=0; i<cbx.length; i++)
    {
        if (cbx.options[i].value == hdn.value)
        {
            cbx.options[i].selected = true
        }
    }
} //fim for
} //fim fuction

var carregar = 0;
function verificaOpcao(Id,Arquivo,Size)
{
    a = document.getElementById(Id);
    if(a.options[a.selectedIndex].value == "novoCampo")
    {
        Size = Size.split(",");
        window.open(Arquivo,"", 'status=1,scrollbars = yes,width='+Size[0]+' ,height='+Size[1]);
        carregar = 1;
    }
}

function createXMLHTTP() {
    try {
        objetus = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            objetus= new ActiveXObject("Microsoft.XMLHTTP");
        } catch (E) {

```

```

        objetus= false;
    }
}
if (!objetus && typeof XMLHttpRequest!='undefined') {
    objetus = new XMLHttpRequest();
}
return objetus
}

function carrega(cel,cbx,Arquivo,Size)
{
    if(carregar == 1){
        var oHTTPRequest = createXMLHTTP();
        oHTTPRequest.open("post", "scripts/combobox.php?cbx="+cbx, true);
        oHTTPRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        oHTTPRequest.onreadystatechange=function() {
            if (oHTTPRequest.readyState==4){
                document.getElementById(cel).innerHTML= "<select name='"+cbx+"'
id='"+cbx+"' onChange='\"verificaOpcao(\"'+cbx+'\",'+Arquivo+'\",'+Size+'\"'\)\"
onFocus='\"carrega(\"'+cel+'\",'+cbx+'\",'+Arquivo+'\",'+Size+'\"'\)\">"+oHTTPRequest.responseText+"<o
ption value='novoCampo'>Novo...</option></select>";
            }
        }
        oHTTPRequest.send();
        carregar = 0;
    }
}
</script>

```

```

<td valign="top">
<form name="cadTela" method="post" action="constroi_Entidade.php?Salva=<?echo
$opcao;?>&amp;tela=<?echo $_REQUEST["tela"];?>" onSubmit="return fValidaCampos()">
<table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos2">
<tr bgcolor="#339933">
<th height="24" colspan="2" bordercolor="#D4D0C8" bgcolor="#339933" id="titulo"><div
align="left"><font color="#FFFFFF" size="3" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Montagem de Entidade </strong></font></div></th>
<th width="22%" bgcolor="#339933"><div align="right"><font size="2"><a
href="javascript:history.back()" target="_top"><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><strong><font face="Verdana, Arial, Helvetica, sans-serif">
<u>Voltar</u></font></strong></font></a></font>&nbsp;&nbsp;&nbsp;</div></th>
</tr>
<tr>
<td width="22%" height="57"><p>
<input type="hidden" name="tpTela" value="Entidade">
<input type="hidden" name="hdnDados" id="hDados">
<input type="hidden" name="hdnFases" id="hFases">
</p>
<p><strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">NOME DA
ENTIDADE:</font></strong></p></td>
<td width="56%"><p>
<input name="edtNome" type="text" size="28" value="<?echo $tela->getNome();?>">
</p>
</td>

```



```

$res = $tela->excluir($_REQUEST['tela']);

if ($res == true){?>
<script language='JavaScript'>
    window.alert("Entidade excluída com sucesso!");
    window.parent.location = '../bancoentidade.php';
</script>
<?
}
?>

```

Nome do arquivo: bancofases.php

Descrição: código que gera a Lista de fases, chamando a função fTransformaXML() da classe TrasformaXML.

```

<?
    include ("../classes/clsConverteXML.php");

    if(file_exists("../xml/bancofases.xml")){
        $txml = new clsConverte();
        $txml->fTransformaXML("../xml/lista_Fase.xml","../xml/bancofases.xml");
    }else{
        echo "<br><br><center><font color=#339933 face=Verdana, Arial, Helvetica, sans-
serif><b>N&atilde;o h&aacute; fases cadastradas.</b><p><a href='constroi_Fase.php'>Cadastrar nova
fase</p></font></center>";
    }
?>

```

Nome do arquivo: lista_Fase.xml

Descrição: código XSL que converte bancofases.xml em uma tabela na Lista de fases.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<title>Lista de telas</title>
</head>
<body>
    <table width="100%" border="0" align="center" cellpadding="1" cellspacing="1" >
        <tr align="left"><th colspan="3" bgcolor="#339933"><font color="#ffffff" face="Verdana, Arial,
Helvetica, sans-serif"> Lista de Fases </font></th>
        </tr>
        <tr><td colspan="3" >
            <a><xsl:attribute name="href">../telas/constroi_Fase.php</xsl:attribute><font color="#339999"
face="Verdana, Arial, Helvetica, sans-serif"><strong><u>Nova</u></strong></font></a><br /><br
/></td></tr>

```

```

<xsl:for-each select="banco_telas/FASE">
  <xsl:if test="@ATIVO='S'">
    <tr bgcolor="66CC66">
      <td width="80%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif"><b><xsl:value-of select="@nome"/></b></font></td>
      <td align="center">
        <a><xsl:attribute name="href">../telas/constroi_Fase.php?tela=<xsl:value-of
select="@CODIGO"/></xsl:attribute>
        <font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Editar</b></font>
        </a></td>
      <td align="center">
        <a><xsl:attribute name="href">../telas/scripts/ExcluiFase.php?tela=<xsl:value-of
select="@CODIGO"/></xsl:attribute>
        <font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Apagar</b></font>
        </a></td>
      </tr>
    </xsl:if>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Nome do arquivo: constroi_Fase.php

Descrição: código que monta a tela de cadastro e edição de acervo de Fase. Quando o usuário clica em salvar, recebe os dados e, de acordo com a opção enviada, chama a função fSalvaTela() ou editar() da classe Acervo (tela).

```

<?php
  include("login.inc");
  include_once('../classes/clsCampos.php');
  include_once('../classes/clsTela.php');
  if($_SESSION['tipo']!="Administrador"){?>
  <script language="javascript">
    document.location = "principal_user.php";
  </script>
<? }
  $opcao="S";
  $tela = new clsTela();
  if ($_REQUEST["Salva"]=="S"){
    $tela->SetXML("../xml/bancofases.xml");
    $tela->SetTipo($_POST["tpTela"]);
    $tela->SetCampos($_POST["hdnDados"]);
    $tela->SetMetadado($_POST["hdnProcessos"]);
    $tela->SetNome($_POST["edtNome"]);
    if ($tela->fSalvaTela()){
      ?>
    <script language="JavaScript">

```

```

        alert("Tela gravada com sucesso!");
        document.location = "bancofases.php";
    </script>
    <?
    }
}elseif($_REQUEST["Salva"]=="E"){
    $tela->SetXML("../xml/bancofases.xml");
    $tela->SetTipo($_POST["tpTela"]);
    $tela->SetCampos($_POST["hdnDados"]);
    $tela->SetMetadado($_POST["hdnProcessos"]);
    $tela->SetNome($_POST["edtNome"]);
    if ($tela->editar($_REQUEST["tela"])){
        ?>
        <script language="JavaScript">
            alert("Tela alterada com sucesso!");
            document.location = "bancofases.php";
        </script>
        <?
        }
}elseif($_REQUEST["tela"]!=""){
    $tela->SetXML("../xml/bancofases.xml");
    $tela->setTela($_REQUEST["tela"]);
    $opcao="E";
}
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Sistema rastreabilidade</title>
<script language="JavaScript" type="text/javascript">
    window.defaultStatus=".: Sistema de Rastreabilidade";
    document.onmousedown=verfonte;
    function verfonte()
    {
        if (event.button==2)
        {
            // window.alert('Todos os direitos reservados - Ana e Eduardo')
        }
    } //fim ver fonte
    vetCampos = new Array(); //variável que será usada para gravar os dados no banco
    vetProcessos = new Array(); //variável que será usada para gravar as fases no banco
    <?
        $codCampos = explode(",",$tela->getCampos());
        $cont = count($codCampos);
        for ($x=0;$x<$cont-1;$x++){
            echo "add = vetCampos.splice($x,0,$codCampos[$x]); \n";
        }
        if($tela->GetMetadado()!=""){
            $codMetadado = explode(",",$tela->GetMetadado());
            $cont = count($codMetadado);
            for ($x=0;$x<$cont;$x++){
                echo "add = vetProcessos.splice($x,0,$codMetadado[$x]); \n";
            }
        }
    }
}

```

```

?>
cont=-1; //cont recebe -1 para ser usado abaixo
function adCampo(index) //index serve para indicar a posiç o do campo no vetor que deve ser o
mesmo do select
{
    var i=document.getElementById('ordem').value // captura o valor do campo ordem para inserir o
registro na ordem indicada
    var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
    var a=document.getElementById("cbxCampos") //captura o select cbxlista p/ uma vari vel a
        var dados= a.options[a.selectedIndex].value.split("|") //transforma um campo de aTphHtml em
um novo array chamado dados
    add = vetCampos.splice(i-1,0,dados[0])
    var y=x.insertCell(0) //insere as quatro c lulas da tabela
    var z=x.insertCell(1)
    var v=x.insertCell(2)
    y.innerHTML=(a.options[a.selectedIndex].text) //altera o valor do html colocando o valor do texto
do item selecionado no select
        if (dados[2] == 'checkbox'){
            var opcoes= dados[3].split(";")
            var cont=0
            for (cont;cont<opcoes.length;cont++){
                z.innerHTML=z.innerHTML+"<input type=checkbox name="+opcoes[cont]+
value=checkbox disabled>"+opcoes[cont]+<p>"
            }
        }
        if (dados[2] == 'textarea'){
            z.innerHTML="<textarea name="+dados[1]+<input type=checkbox name="+opcoes[cont]+
value=checkbox disabled>"+opcoes[cont]+<p>"
        }
        if (dados[2] == 'select'){
            var opcoes= dados[3].split(";")
            var cont=0
            var vselect
                vselect="<select name=select1>"
            for (cont;cont<opcoes.length;cont++){
                vselect=vselect+"<option
value="+opcoes[cont]+>"+opcoes[cont]+</option>"
            }
            z.innerHTML=vselect+"</select>"
        }
        if (dados[2] == 'text') {
            z.innerHTML="<input type='text' name="+dados[1]+<input type=checkbox name="+opcoes[cont]+
value=checkbox disabled>"+opcoes[cont]+<p>"
        }
        v.innerHTML="<a href=# onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
document.getElementById('ordem').value++
    }

function edCampo() //index serve para indicar a posiç o do campo no vetor que deve ser o
mesmo do select
{
    var cbx = document.getElementById("cbxCampos");
    var a;
    for(a=0;a<=vetCampos.length-1;a++){ //for 01
        var b=0;

```

```

        for(b;b<cbx.length;b++){
            var teste= cbx[b].value.split("|"); //transforma um campo de aTpHtml em um novo
array chamado teste
            if(teste[0]==vetCampos[a]){
                dados = cbx[b].value.split("|");
            }
        }

        var i= document.getElementById('ordem').value; // captura o valor do campo ordem
para inserir o registro na ordem indicada
        var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
        var y=x.insertCell(0) //insere as quatro células da tabela
        var z=x.insertCell(1)
        var v=x.insertCell(2)
        y.innerHTML=(dados[1]) //altera o valor do html colocando o valor do texto do item
selecionado no select
        if (dados[2] == 'checkbox'){

            var opcoes= dados[3].split(";")
            var cont=0
            for (cont;cont<opcoes.length;cont++){
                v.innerHTML=v.innerHTML+"<input type=checkbox name="+opcoes[cont]+
value=checkbox disabled>"+opcoes[cont]+<p>"
            }
        }
        if (dados[2] == 'textarea'){
            v.innerHTML="<textarea name="+dados[1]+ " disabled></textarea>"
        }
        if (dados[2] == 'select'){
            var opcoes= dados[3].split(";")
            var cont=0
            var vselect
            vselect="<select name=select1>"
            for (cont;cont<opcoes.length;cont++){
                vselect=vselect+"<option value="+opcoes[cont]+ ">"+opcoes[cont]+</option>"
            }
            v.innerHTML=vselect+"</select>"
        }

        if (dados[2] == 'text') {
            v.innerHTML="<input type='text' name="+dados[1]+ " disabled>"
        }
        z.innerHTML="<a href=#
onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
        document.getElementById('ordem').value++

    } //fim for 01
}
var i =0;
function adMetadado() //index serve para indicar a posição do campo no vetor que deve ser o
mesmo do select
{
    var x=document.getElementById('tblProcessos').insertRow(i+1); //insere a linha na tabela campos
    var a=document.getElementById("cbxProcessos"); //captura o select cbxlista p/ uma variável a

```

```

    add = vetProcessos.splice(i,0,a.value);
    var y=x.insertCell(0); //insere as quatro células da tabela
    var v=x.insertCell(1);
    y.innerHTML=(a.options[a.selectedIndex].text); //altera o valor do html colocando o valor do texto
do item selecionado no select
    v.innerHTML="<a href=#
onclick='deleteMeta(this.parentNode.parentNode.rowIndex)'>Excluir</a>";
    i++;
}

function edMetadado() //index serve para indicar a posição do campo no vetor que deve ser o
mesmo do select
{
    var c=document.getElementById("cbxProcessos"); //captura o select cbxlista p/ uma variável a
    var a;
    for(a=0;a<=vetProcessos.length-1;a++){ //for 01
        for(b=0;b<c.length-1;b++){
            if(vetProcessos[a]==c.options[b].value){
                var nome=c.options[b].text;
            }
        }
        var x=document.getElementById('tblProcessos').insertRow(i+1); //insere a linha na tabela
campos
        var y=x.insertCell(0); //insere as quatro células da tabela
        var v=x.insertCell(1)
        y.innerHTML=nome; //altera o valor do html colocando o valor do texto do item selecionado
no select
        v.innerHTML="<a href=#
onclick='deleteMeta(this.parentNode.parentNode.rowIndex)'>Excluir</a>";
        i++;
    }
}

function deleteRow(i)
{
    document.getElementById('campos').deleteRow(i)
    document.getElementById('ordem').value--
    de_fora = vetCampos.splice(i-1,1)
    //alert(vetCampos)
}

function deleteMeta(index)
{
    document.getElementById('tblProcessos').deleteRow(index)
    de_fora = vetProcessos.splice(index-1,1)
    i--;
    //alert(vetProcessos) ;
}

//função para colocar o array numa variável antes de submeter
function enviaarray()
{
    document.getElementById('hDados').value=vetCampos;
    document.getElementById('hProcessos').value=vetProcessos;
}

```

```

}

function fValidaCampos()
{
    if(document.cadTela.edtNome.value.length == 0)
    {
        window.alert("O Campo 'Nome' precisa ser preenchido");
        document.cadTela.edtNome.focus();
        return false;
    }
}

var carregar = 0;
function verificaOpcao(Id,Arquivo,Size)
{
    a = document.getElementById(Id);
    if(a.options[a.selectedIndex].value == "novoCampo")
    {
        Size = Size.split(",");
        window.open(Arquivo,"", 'status=1,scrollbars = yes,width='+Size[0]+' ,height='+Size[1]);
        carregar = 1;
    }
}

function createXMLHTTP() {
    try {
        objetus = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            objetus= new ActiveXObject("Microsoft.XMLHTTP");
        } catch (E) {
            objetus= false;
        }
    }
    if (!objetus && typeof XMLHttpRequest!='undefined') {
        objetus = new XMLHttpRequest();
    }
    return objetus
}

function carrega(ceI,cbx,Arquivo,Size)
{
    if(carregar == 1){
        var oHttpRequest = createXMLHTTP();
        oHttpRequest.open("post", "scripts/combobox.php?cbx="+cbx, true);
        oHttpRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        oHttpRequest.onreadystatechange=function() {
            if (oHttpRequest.readyState==4){
                document.getElementById(ceI).innerHTML= "<select name='"+cbx+"'
id='"+cbx+"' on change='"+verificaOpcao('\'+cbx+"\',\'+Arquivo+'\',\'+Size+'\'\'
onFocus='"+carrega('\'+ceI+'\',\'+cbx+'\',\'+Arquivo+'\',\'+Size+'\'\'")>'+oHttpRequest.responseText+"<option value='novoCampo'>Novo...</option></select>";
            }
        }
    }
}

```

```

    }
    }
    oHTTPRequest.send();
    carregar = 0;
}
}
</script>
<form name="cadTela" method="post" action="constroi_Fase.php?Salva=<?echo
$opcao;?&telas=<?echo $_REQUEST["tela"];?>" onSubmit="return fValidaCampos()">
  <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos2">
    <tr bgcolor="#339933">
      <th height="22" colspan="2" bgcolor="#339933" id="titulo"><div align="left"><font color="#FFFFFF"
size="3" face="Verdana, Arial, Helvetica, sans-serif">Montagem de Fase</font></div></th>
      <th width="22%"><div align="right"><font size="2"><a href="javascript:history.back()"
target="_top"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><strong><font face="Verdana,
Arial, Helvetica, sans-serif"><u>Voltar</u></font></strong></font></a></font>&nbsp;&nbsp;&nbsp;</div></th>
    </tr>
  </table>
  <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos2">
    <tr>
      <td width="22%" height="67"><p>
        <input type="hidden" name="tpTela" value="Fase">
        <input type="hidden" name="hdnDados" id="hDados">
        <input type="hidden" name="hdnProcessos" id="hProcessos">
      </p>
      <p><strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">NOME DA
FASE:</font></strong></p></td>
      <td width="49%">
        <p><input name="edtNome" type="text" size="28" value="<?echo $tela->getNome();?>"></p>
      </td>
      <td width="29%" align="right"><input type="submit" name="Submit2" value="Salvar"
onClick="enviaarray()"></td>
    </tr>
  </table>
  <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos3">
    <tr>
      <td width="12%" height="69">
        <p align="justify"><font size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Campo:</strong></font>
      </td>
      <td width="24%" id="lista_campos">
        <select name="cbxCampos" id="cbxCampos"
onChange="verificaOpcao('cbxCampos','CampoNovo.php','300,200')"
onFocus="carrega('lista_campos','cbxCampos','CampoNovo.php','300,200')">
          <?
            $campo = new clsCampos(null,'../xml/bancocampo.xml');
            echo $campo->fMontaLista();
          ?>
          <option value="novoCampo">Novo...</option>
        </select>
      </td>
    </tr>
  </table>

```


Nome do arquivo: ExcluiFase.php

Descrição: script que chama a função `excluir()` da classe `Acervo` quando o usuário clica no link Apagar da Lista de fases.

```
<?
include_once('../classes/clsTela.php');
//Código da Página.
$tela = new clsTela();
$tela->SetXML('../xml/bancofases.xml');
$tela->SetTipo("FASE");
$res = $tela->excluir($_REQUEST['tela']);
if ($res == true){?>
<script language='JavaScript'>
    window.alert("Fase excluída com sucesso!");
    window.parent.location = '../bancofases.php';
</script>
<?
}
?>
```

Nome do arquivo: bancoorigem.php

Descrição: código que gera a Lista de origens, chamando a função `fTransformaXML()` da classe `TrasformaXML`.

```
<?
    include ("../classes/clsConverteXML.php");

    if(file_exists("../xml/bancoorigem.xml")){
        $txml = new clsConverte();
        $txml->fTransformaXML("../xml/lista_Origem.xml","../xml/bancoorigem.xml");
    }else{
        echo "<br><br><center><font color=#339933 face=Verdana, Arial, Helvetica, sans-
serif><b>N&atilde;o h&aacute; origens cadastradas.</b><p><a href='constroi_Origem.php'>Cadastrar
nova origem</p></font></center>";
    }
?>
```

Nome do arquivo: lista_Origem.xsl

Descrição: código XSL que converte bancoorigem.xml em uma tabela na Lista de origens.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<title>Lista de telas</title>
</head>
<body>
  <table width="100%" border="0" align="center" cellpadding="1" cellspacing="1" >
    <tr align="left"><th colspan="3" bgcolor="#339933"><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif">Lista de Origens </font></th>
    </tr>
    <tr><td colspan="3" >
      <a><xsl:attribute name="href">../telas/constroi_Origem.php</xsl:attribute><font color="#339999" face="Verdana, Arial, Helvetica, sans-serif"><strong><u>Nova</u></strong></font></a><br /></td></tr>
      <xsl:for-each select="banco_telas/ORIGEM">
        <xsl:if test="@ATIVO='S'">
          <tr bgcolor="66CC66">
            <td width="80%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:value-of select="@nome"/></b></font></td>
            <td align="center">
              <a><xsl:attribute name="href">../telas/constroi_Origem.php?tela=<xsl:value-of select="@CODIGO"/></xsl:attribute>
                <font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Editar</b></font>
              </a></td>
            <td align="center">
              <a><xsl:attribute name="href">../telas/scripts/ExcluiOrigem.php?tela=<xsl:value-of select="@CODIGO"/></xsl:attribute>
                <font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Apagar</b></font>
              </a></td>
            </tr>
          </xsl:if>
        </xsl:for-each>
      </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Nome do arquivo: constroi_Origem.php

Descrição: código que monta a tela de cadastro e edição de acervo de Origem. Quando o usuário clica em salvar, recebe os dados e, de acordo com a opção enviada, chama a função fSalvaTela() ou editar() da classe Acervo (tela).

```
<?php
    include("login.inc");
    include_once('../classes/clsCampos.php');
    include_once('../classes/clsTela.php');
    if($_SESSION['tipo']!="Administrador")
    {?>
    <script language="javascript">
        document.location = "principal_user.php";
    </script>
    <? }
    $opcao="S";
    $tela = new clsTela();
    if ($_REQUEST["Salva"]=="S"){
        $tela->SetXML("../xml/bancoorigem.xml");
        $tela->SetTipo($_POST["tpTela"]);
        $tela->SetCampos($_POST["hdnDados"]);
        $tela->SetNome($_POST["edtNome"]);
        if ($tela->fSalvaTela()){
            ?>
            <script language="JavaScript">
                alert("Tela gravada com sucesso!");
                document.location ="bancoorigem.php";
            </script>
            <?
            }
        }elseif($_REQUEST["Salva"]=="E"){
            $tela->SetXML("../xml/bancoorigem.xml");
            $tela->SetTipo($_POST["tpTela"]);
            $tela->SetCampos($_POST["hdnDados"]);
            $tela->SetNome($_POST["edtNome"]);
            if ($tela->editar($_REQUEST["tela"])){
                ?>
                <script language="JavaScript">
                    alert("Tela alterada com sucesso!");
                    document.location ="bancoorigem.php";
                </script>
                <?
                }
            }elseif($_REQUEST["tela"]!=""){
                $tela->SetXML("../xml/bancoorigem.xml");
                $tela->setTela($_REQUEST["tela"]);
                $opcao="E";
            }
        }
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```

<html>
<head>
<title>Sistema rastreabilidade</title>
<script language="JavaScript" type="text/javascript">
    window.defaultStatus=".: Sistema de Rastreabilidade";
    document.onmousedown=verfonte;
    function verfonte()
    {
        if (event.button==2)
        {
            window.alert('Todos os direitos reservados - Ana e Eduardo')
        }
    } //fim ver fonte
    vetCampos = new Array(); //variável que será usada para gravar os dados no banco
    <?
        $codCampos = explode(",",$tela->getCampos());
        $cont = count($codCampos);
        for ($x=0;$x<$cont-1;$x++){
            echo "add = vetCampos.splice($x,0,$codCampos[$x]); \n";
        }
    ?>
    cont=-1; //cont recebe -1 para ser usado abaixo
    function adCampo(index) //index serve para indicar a posição do campo no vetor que deve ser o
    mesmo do select
    {
        var i=document.getElementById('ordem').value // captura o valor do campo ordem para inserir o
        registro na ordem indicada
        var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
        var a=document.getElementById("cbxCampos") //captura o select cbxlista p/ uma variável a
        var dados= a.options[a.selectedIndex].value.split("|") //transforma um campo de aTpHtml em
        um novo array chamado dados
        add = vetCampos.splice(i-1,0,dados[0])
        var y=x.insertCell(0) //insere as quatro células da tabela
        var z=x.insertCell(1)
        var v=x.insertCell(2)
        y.innerHTML=(a.options[a.selectedIndex].text) //altera o valor do html colocando o valor do texto
        do item selecionado no select
        if (dados[2] == 'checkbox'){
            var opcoes= dados[3].split(";")
            var cont=0
            for (cont;cont<opcoes.length;cont++){
                z.innerHTML=z.innerHTML+"<input type=checkbox name="+opcoes[cont]+
                "value=checkbox disabled>"+opcoes[cont]+<p>"
            }
        }
        if (dados[2] == 'textarea'){
            z.innerHTML="<textarea name="+dados[1]+ " disabled>"+dados[3]+</textarea>"
        }
        if (dados[2] == 'select'){
            var opcoes= dados[3].split(";")
            var cont=0
            var vselect
            vselect="<select name=select1>"

```

```

                for (cont;cont<opcoes.length;cont++){
                    vselect=vselect+"<option
value="+opcoes[cont]+>" +opcoes[cont]+</option>"
                }
                z.innerHTML=vselect+"</select>"
            }
            if (dados[2] == 'text') {
                z.innerHTML="<input type='text' name="+dados[1]+> disabled value="+dados[3]+>"
            }
            v.innerHTML="<a href=# onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
            document.getElementById('ordem').value++
        }
        function edCampo() //index serve para indicar a posiç o do campo no vetor que deve ser o
mesmo do select
        {
            var cbx = document.getElementById("cbxCampos");
            var a;
            for(a=0;a<=vetCampos.length-1;a++){ //for 01
                var b=0;
                for(b; b<cbx.length;b++){
                    var teste= cbx[b].value.split("|"); //transforma um campo de aTphTml em um novo
array chamado teste
                    if(teste[0]==vetCampos[a]){
                        dados = cbx[b].value.split("|");
                    }
                }
                var i= document.getElementById('ordem').value; // captura o valor do campo ordem
para inserir o registro na ordem indicada
                var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
                var y=x.insertCell(0) //insere as quatro c lulas da tabela
                var z=x.insertCell(1)
                var v=x.insertCell(2)
                y.innerHTML=(dados[1]) //altera o valor do html colocando o valor do texto do item
selecionado no select
                if (dados[2] == 'checkbox'){
                    var opcoes= dados[3].split(";")
                    var cont=0
                    for (cont;cont<opcoes.length;cont++){
                        v.innerHTML=v.innerHTML+"<input type=checkbox name="+opcoes[cont]+>"
value=checkbox disabled">"+opcoes[cont]+<p>"
                    }
                }
                if (dados[2] == 'textarea'){
                    v.innerHTML="<textarea name="+dados[1]+> disabled</textarea>"
                }
                if (dados[2] == 'select'){
                    var opcoes= dados[3].split(";")
                    var cont=0
                    var vselect
                    vselect="<select name=select1>"
                    for (cont;cont<opcoes.length;cont++){
                        vselect=vselect+"<option value="+opcoes[cont]+>" +opcoes[cont]+</option>"
                    }
                    v.innerHTML=vselect+"</select>"
                }
            }
        }

```

```

        }
        if (dados[2] == 'text') {
            v.innerHTML="<input type='text' name="+dados[1]+" disabled>"
        }
        z.innerHTML="<a href=#
onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
        document.getElementById('ordem').value++
    }//fim for 01
}

function deleteRow(i)
{
    document.getElementById('campos').deleteRow(i)
    document.getElementById('ordem').value--
    de_fora = vetCampos.splice(i-1,1)
    //alert(vetCampos)
}

//função para colocar o array numa variável;vel antes de submeter
function enviaarray()
{
    document.getElementById('hDados').value=vetCampos;
}

function fValidaCampos()
{
    if(document.cadTela.edtNome.value.length == 0)
    {
        window.alert("O Campo 'Nome' precisa ser preenchido");
        document.cadTela.edtNome.focus();
        return false;
    }
}

var carregar = 0;
function verificaOpcao(Id,Arquivo)
{
    a = document.getElementById(Id);
    if(a.options[a.selectedIndex].value == "novoCampo")
    {
        window.open(Arquivo,"", 'status=1,width=300,height=200');
        carregar = 1;
    }
}

function createXMLHTTP() {
    try {
        objetus = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            objetus= new ActiveXObject("Microsoft.XMLHTTP");
        } catch (E) {

```

```

        objetus= false;
    }
}
if (!objetus && typeof XMLHttpRequest!='undefined') {
    objetus = new XMLHttpRequest();
}
return objetus
}

function carrega(cel,cbx,Arquivo)
{
    if(carregar == 1){
        var oHTTPRequest = createXMLHTTP();
        oHTTPRequest.open("post", "scripts/combobox.php?cbx="+cbx, true);
        oHTTPRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        oHTTPRequest.onreadystatechange=function() {
            if (oHTTPRequest.readyState==4){
                document.getElementById(cel).innerHTML= "<select name=\""+cbx+"\"
id=\""+cbx+"\" onChange=\"\`verificaOpcao(\""+cbx+"\",\""+Arquivo+"\")\"
onFocus=\"\`carrega(\""+cel+"\",\""+cbx+"\",\""+Arquivo+"\")\">"+oHTTPRequest.responseText+"<option
value='novoCampo'>Novo...</option></select>";
            }
        }
        oHTTPRequest.send();
        carregar = 0;
    }
}
</script>
<form name="cadTela" method="post" action="constroi_Origem.php?Salva=<?echo
$opcao;?>&amp;tela=<?echo $_REQUEST["tela"];?>" onSubmit="return fValidaCampos()">
    <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos2">
        <tr bgcolor="#339933">
            <th height="22" colspan="2" bgcolor="#339933" id="titulo"><div align="left"><font
color="#FFFFFF" size="3" face="Verdana, Arial, Helvetica, sans-serif">Montagem de
Origem</font></div></th>
            <th width="22%"><div align="right"><font size="2"><a href="javascript:history.back()"
target="_top"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><strong><font face="Verdana,
Arial, Helvetica, sans-serif"><u>Voltar</u></font></strong></font></a></font>&nbsp;&nbsp;&nbsp;</div></th>
        </tr>
        <tr>
            <td height="63"><p>
                <input type="hidden" name="tpTela" value="Origem">
                <input type="hidden" name="hdnDados" id="hDados"></p>
                <p><strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO
                DA ORIGEM:</font></strong></p>
            </td>
            <td width="57%"><p>
                <input name="edtNome" type="text" size="28" value="<?echo $tela-
>getNome();?>">
            </p>
            </td>
            <td align="right"><input type="submit" name="Submit2" value="Salvar"
onClick="enviaarray()"></td>
        </tr>
    </table>

```


Nome do arquivo: ExcluiOrigem.php

Descrição: script que chama a função `excluir()` da classe `Acervo` quando o usuário clica no link Apagar da Lista de origens.

```
<?
include_once('../classes/clsTela.php');
//Código da Página.
$tela = new clsTela();
$tela->SetXML('../xml/bancoorigem.xml');
$tela->SetTipo("ORIGEM");
$res = $tela->excluir($_REQUEST['tela']);
if ($res == true){?>
<script language='JavaScript'>
    window.alert("Origem excluída com sucesso!");
    window.parent.location = '../bancoorigem.php';
</script>
<?
}
?>
```

Nome do arquivo: bancoprocessos.php

Descrição: código que gera a Lista de processos, chamando a função `fTransformaXML()` da classe `TrasformaXML`.

```
<?
    include ("../classes/clsConverteXML.php");
    if(file_exists("../xml/bancoprocessos.xml")){
        $txml = new clsConverte();
        $txml->fTransformaXML("../xml/lista_Processo.xml","../xml/bancoprocessos.xml");
    }else{
        echo "<br><br><center><font color=#339933 face=Verdana, Arial, Helvetica, sans-
serif><b>N&atilde;o h&aacute; processos cadastrados.</b><p><a
href='constroi_processo.php'>Cadastrar novo processo</p></font></center>";
    }
?>
```

Nome do arquivo: lista_Processo.xsl

Descrição: código XSL que converte bancoprocessos.xml em uma tabela na Lista de processos.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<title>Lista de telas</title>
</head>
<body>
<table width="100%" border="0" align="center" cellpadding="1" cellspacing="1" >
<tr align="left"><th colspan="3" bgcolor="#339933"><font color="#ffffff" face="Verdana, Arial,
Helvetica, sans-serif">Lista de Processos </font></th>
</tr>
<tr><td colspan="3" >
<a><xsl:attribute name="href">../telas/constroi_processo.php</xsl:attribute><font
color="#339999" face="Verdana, Arial, Helvetica, sans-
serif"><strong><u>Novo</u></strong></font></a><br /></td></tr>
<xsl:for-each select="banco_telas/PROCESSO">
<xsl:if test="@ATIVO='S'">
<tr bgcolor="66CC66">
<td width="80%"><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif"><b><xsl:value-of select="@nome"/></b></font></td>
<td align="center">
<a><xsl:attribute name="href">../telas/constroi_processo.php?tela=<xsl:value-of
select="@CODIGO"/></xsl:attribute>
<font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Editar</b></font>
</a></td>
<td align="center">
<a><xsl:attribute name="href">../telas/scripts/ExcluiProcesso.php?tela=<xsl:value-of
select="@CODIGO"/></xsl:attribute>
<font color="#006600" face="Verdana, Arial, Helvetica, sans-serif"><b>Apagar</b></font>
</a></td>
</tr>
</xsl:if>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Nome do arquivo: constroi_processo.php

Descrição: código que monta a tela de cadastro e edição de acervo de Processo. Quando o usuário clica em salvar, recebe os dados e, de acordo com a opção enviada, chama a função fSalvaTela() ou editar() da classe Acervo (tela).

```
<?php
    include("login.inc");
    include_once("../classes/clsCampos.php");
    include_once("../classes/clsTela.php");
    if($_SESSION['tipo']!="Administrador"){?>
    <script language="javascript">
        document.location = "principal_user.php";
    </script>
<? }
    $opcao="S";
    $tela = new clsTela();
    if ($_REQUEST["Salva"]=="S"){
        $tela->SetXML("../xml/bancoprocessos.xml");
        $tela->SetTipo($_POST["tpTela"]);
        $tela->SetCampos($_POST["hdnDados"]);
        $tela->SetNome($_POST["edtNome"]);
        if ($tela->fSalvaTela()){
            ?>
            <script language="JavaScript">
                alert("Tela gravada com sucesso!");
                document.location = "bancoprocessos.php";
            </script>
            <?
            }
        }elseif($_REQUEST["Salva"]=="E"){
            $tela->SetXML("../xml/bancoprocessos.xml");
            $tela->SetTipo($_POST["tpTela"]);
            $tela->SetCampos($_POST["hdnDados"]);
            $tela->SetNome($_POST["edtNome"]);
            if ($tela->editar($_REQUEST["tela"])){
                ?>
                <script language="JavaScript">
                    alert("Tela alterada com sucesso!");
                    document.location = "bancoprocessos.php";
                </script>
                <?
                }
            }elseif($_REQUEST["tela"]!=""){
                $tela->SetXML("../xml/bancoprocessos.xml");
                $tela->setTela($_REQUEST["tela"]);
                $opcao="E";
            }
        }
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
```

```

<head>
<title>Sistema rastreabilidade</title>
<script language="JavaScript" type="text/javascript">
    window.defaultStatus=".: Sistema de Rastreabilidade";
    document.onmousedown=verfonte;
    function verfonte()
    {
        if (event.button==2)
            {
                window.alert('Todos os direitos reservados - Ana e Eduardo')
            }
    } //fim ver fonte

    vetCampos = new Array(); //variável que será usada para gravar os dados no banco

<?
    $codCampos = explode(",",$tela->getCampos());
    $cont = count($codCampos);
    for ($x=0;$x<$cont-1;$x++){
        echo "add = vetCampos.splice($x,0,$codCampos[$x]); \n";
    }

?>

    cont=-1; //cont recebe -1 para ser usado abaixo
    function adCampo(index) //index serve para indicar a posição do campo no vetor que deve ser o
    mesmo do select
    {
        var i=document.getElementById('ordem').value // captura o valor do campo ordem para inserir o
        registro na ordem indicada
        var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
        var a=document.getElementById("cbxCampos") //captura o select cbxlista p/ uma variável a
        var dados= a.options[a.selectedIndex].value.split("|") //transforma um campo de aTpHtml em
        um novo array chamado dados
        add = vetCampos.splice(i-1,0,dados[0])
        var y=x.insertCell(0) //insere as quatro células da tabela
        var z=x.insertCell(1)
        var v=x.insertCell(2)
        y.innerHTML=(a.options[a.selectedIndex].text) //altera o valor do html colocando o valor do texto
        do item selecionado no select
        if (dados[2] =='checkbox'){
            var opcoes= dados[3].split(";")
            var cont=0
            for (cont;cont<opcoes.length;cont++){
                z.innerHTML=z.innerHTML+"<input type=checkbox name="+opcoes[cont]+
                value=checkbox disabled>"+opcoes[cont]+<p>"
            }
        }
        if (dados[2] == 'textarea'){
            z.innerHTML="<textarea name="+dados[1]+<div data-bbox="875 934 916 952" data-label="Page-Footer">


158


```

```

        var cont=0
        var vselect
            vselect="<select name=select1>"
        for (cont;cont<opcoes.length;cont++){
            vselect=vselect+"<option
value="+opcoes[cont]+>">"+opcoes[cont]+</option>"
            }
        z.innerHTML=vselect+"</select>"
    }
    if (dados[2] == 'text') {
        z.innerHTML="<input type='text' name="+dados[1]+> disabled value="+dados[3]+>"
    }
    v.innerHTML="<a href=# onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
    document.getElementById('ordem').value++
}

function edCampo() //index serve para indicar a posiÃ§Ã£o do campo no vetor que deve ser o
mesmo do select
{
    var cbx = document.getElementById("cbxCampos");
    var a;
    for(a=0;a<=vetCampos.length-1;a++){ //for 01
        var b=0;
        for(b;b<cbx.length;b++){
            var teste= cbx[b].value.split("|"); //transforma um campo de aTpHtml em um novo
array chamado teste
            if(teste[0]==vetCampos[a]){
                dados = cbx[b].value.split("|");
            }
        }
    }
    var i= document.getElementById('ordem').value; // captura o valor do campo ordem
para inserir o registro na ordem indicada
    var x=document.getElementById('campos').insertRow(i)//insere a linha na tabela campos
    var y=x.insertCell(0) //insere as quatro cÃ©lulas da tabela
    var z=x.insertCell(1)
    var v=x.insertCell(2)
    y.innerHTML=(dados[1]) //altera o valor do html colocando o valor do texto do item
selecionado no select
    if (dados[2] == 'checkbox'){
        var opcoes= dados[3].split(";")
        var cont=0
        for (cont;cont<opcoes.length;cont++){
            v.innerHTML=v.innerHTML+"<input type=checkbox name="+opcoes[cont]+>"
value=checkbox disabled">"+opcoes[cont]+<p>"
        }
    }
    if (dados[2] == 'textarea'){
        v.innerHTML="<textarea name="+dados[1]+> disabled</textarea>"
    }
    if (dados[2] == 'select'){
        var opcoes= dados[3].split(";")
        var cont=0
        var vselect

```

```

        vselect="<select name=select1>"
        for (cont;cont<opcoes.length;cont++){
            vselect=vselect+"<option value="+opcoes[cont]+>"+opcoes[cont]+</option>"
        }
        v.innerHTML=vselect+"</select>"
    }
    if (dados[2] == 'text') {
        v.innerHTML="<input type='text' name="+dados[1]+" disabled>"
    }
    z.innerHTML="<a href=#
onclick='deleteRow(this.parentNode.parentNode.rowIndex)'>Excluir</a>"
    document.getElementById('ordem').value++
} //fim for 01
}

function deleteRow(i)
{
    document.getElementById('campos').deleteRow(i)
    document.getElementById('ordem').value--
    de_fora = vetCampos.splice(i-1,1)
    //alert(vetCampos)
}

//função para colocar o array numa variável antes de submeter
function enviaarray()
{
    document.getElementById('hDados').value=vetCampos;
}

function fValidaCampos()
{
    if(document.cadTela.edtNome.value.length == 0)
    {
        window.alert("O Campo 'Nome' precisa ser preenchido");
        document.cadTela.edtNome.focus();
        return false;
    }
}

var carregar = 0;
function verificaOpcao(Id,Arquivo)
{
    a = document.getElementById(Id);
    if(a.options[a.selectedIndex].value == "novoCampo")
    {
        window.open(Arquivo,"", 'status=1,width=300,height=200');
        carregar = 1;
    }
}

function createXMLHTTP() {
    try {
        objetus = new ActiveXObject("Msxml2.XMLHTTP");
    }
}

```

```

} catch (e) {
    try {
        objetus= new ActiveXObject("Microsoft.XMLHTTP");
    } catch (E) {
        objetus= false;
    }
}
if (!objetus && typeof XMLHttpRequest!='undefined') {
    objetus = new XMLHttpRequest();
}
return objetus
}

function carrega(cel,cbx,Arquivo)
{
    if(carregar == 1){
        var oHTTPRequest = createXMLHTTP();
        oHTTPRequest.open("post", "scripts/combobox.php?cbx="+cbx, true);
        oHTTPRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        oHTTPRequest.onreadystatechange=function() {
            if (oHTTPRequest.readyState==4){
                document.getElementById(cel).innerHTML= "<select name='"+cbx+"\"
id='"+cbx+"\" onChange='\"verificaOpcao(\"'+cbx+"\",'+Arquivo+'\"'\")\"
onFocus='\"carrega(\"'+cel+"\",'+cbx+"\",'+Arquivo+'\"'\")\">"+oHTTPRequest.responseText+"<option
value='novoCampo'>Novo...</option></select>";
            }
        }
        oHTTPRequest.send();
        carregar = 0;
    }
}
</script>
<form name="cadTela" method="post" action="constroi_processo.php?Salva=<?echo
$opcao;?>&amp;tela=<?echo $_REQUEST["tela"];?>" onSubmit="return fValidaCampos()">
    <table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos2">
        <tr bgcolor="#339933">
            <th height="22" colspan="2" bgcolor="#339933" id="titulo"><div align="left"><font
color="#FFFFFF" size="3" face="Verdana, Arial, Helvetica, sans-serif">Montagem de
Processo</font></div></th>
            <th width="22%"><div align="right"><font size="2"><a href="javascript:history.back()"
target="_top"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><strong><font face="Verdana,
Arial, Helvetica, sans-serif"><u>Voltar</u></font></strong></font></a></font>&nbsp;&nbsp;&nbsp;</div></th>
        </tr>
        <tr>
            <td height="69" width="15%"><p>
                <input type="hidden" name="tpTela" value="Processo">
                <input type="hidden" name="hdnDados" id="hDados">
            </p> <p><strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">NOME
DO PROCESSO:</font></strong></p>
            </td>
            <td width="25%"><p><input name="edtNome" type="text" size="28" value="<?echo $tela-
>getNome();?>"></p></td>
            <td align="right"><input type="submit" name="Submit2" value="Salvar"
onClick="enviaarray()"></td>
    </table>

```

```

</tr>
<tr>
  <td colspan="3"><font>&nbsp;</font></td>
</tr>
</table>
<table width="100%" border="0,5" align="center" cellpadding="0" cellspacing="0" id="campos3">
  <tr>
    <td width="10%" height="69">
      <p align="justify"><font size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Campo:</strong></font>
      </td>
    <td width="20%" id="lista_campos">
      <select name="cbxCampos" id="cbxCampos"
onChange="verificaOpcao('cbxCampos','CampoNovo.php')"
onFocus="carrega('lista_campos','cbxCampos','CampoNovo.php')">
        <?
          $campo = new clsCampos(null,'./xml/bancocampo.xml');
          echo $campo->fMontaLista();
        ?>
        <option value="novoCampo">Novo...</option>
      </select>
    </td>
    <td width="70%">
      <font size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>
&nbsp;&nbsp;&nbsp;Ordem:
      <input type="text" name="edtOrdem" size="8" id="ordem" value="1">
      &nbsp;&nbsp;&nbsp;
      <input type="button" name="Adicionar" value="Adicionar"
onClick="adCampo(document.cadTela.cbxCampos.selectedIndex)">
      </strong></font> </p>
    </td>
  </tr>
</table>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0" id="campos">
  <tr bgcolor="#009999">
    <td width="29%" height="24" bgcolor="#009999" id="nm"><div align="left"><font
color="#FFFFFF" size="3" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Campos</strong></font></div></td>
    <td width="62%" id="tp"><div align="left"></div></td>
    <td width="9%"><div align="right"></div> <div align="center"></div></td>
  </tr>
  <script language="JavaScript">edCampo()</script>
</table>
<p>&nbsp;</p>
<p align="center">&nbsp;</p>
</form>

```

Nome do arquivo: ExcluiProcesso.php

Descrição: script que chama a função `excluir()` da classe `Acervo` quando o usuário clica no link Apagar da Lista de processos.

```
<?
include_once('../classes/clsTela.php');
//Código da Página.
$tela = new clsTela();
$tela->SetXML('../xml/bancoprocessos.xml');
$tela->SetTipo("PROCESSO");
$res = $tela->excluir($_REQUEST['tela']);
if ($res == true){?>
<script language='JavaScript'>
    window.alert("Processo excluído com sucesso!");
    window.parent.location = '../bancoprocessos.php';
</script>
<?
}
?>
```

Nome do arquivo: combobox.php

Descrição: código chamado através de um `XMLHttpRequest`, pelas telas de cadastro de Entidade, Fase, Origem e Processo, ele atualiza os combos quando um novo campo é criado durante a inclusão de um acervo.

```
<?
if($_REQUEST['cbx']=="cbxCampos"){
    include_once("../classes/clsCampos.php");
    $campo = new clsCampos("", '../xml/bancocampo.xml');
    echo $campo->fMontaLista();
}else{
    include_once("../classes/clsTela.php");
    $tela = new clsTela();
    if($_REQUEST['cbx']=="cbxOrigem"){
        echo $tela->fMontaCombo("../xml/bancoorigem.xml", "banco_telas", "ORIGEM");
        echo "ola";
    }elseif($_REQUEST['cbx']=="cbxFases"){
        echo $tela->fMontaCombo("../xml/bancofases.xml", "banco_telas", "FASE");
    }elseif($_REQUEST['cbx']=="cbxProcessos"){
        echo $tela->fMontaCombo("../xml/bancoprocessos.xml", "banco_telas", "PROCESSO");
    }
}
?>
```

Nome do arquivo: xmltree.php

Descrição: este código, que na verdade é uma classe, é a parte mais importante do módulo de rastreabilidade, pois é ele que monta o menu de árvore que é utilizado para visualizar as estruturas dos registros de entidades cadastrados. Esta árvore possui links que apontam para o arquivo dados.php e passam os parâmetros necessários para que se mostrem as partes do arquivo desejadas.

```
<?php
class tree {
    var $file = "../xml/bancoentidade.xml"; // Default file name
    var $fontSize = -1; // Default font size
    var $expandAll = 0; // 1 = yes, 0 = no
    var $imgPath = "arvorexml/treeicon/";
    var $imgType = array("leaf" => "t_leaf.gif",
        "tplus" => "t_tplus.gif",
        "cplus" => "t_cplus.gif",
        "tminus" => "t_tminus.gif",
        "cminus" => "t_cminus.gif",
        "mbar" => "t_bar.gif",
        "mtee" => "t_tee.gif",
        "mcorner" => "t_c.gif",
        "Novo" => "mais.gif",
        "Origem" => "t_ini.gif",
        "Entidade" => "t_pdf.gif",
        "Entidades" => "t_web.gif",
        "Fases" => "t_text.gif",
        "Fase" => "t_web1.gif",
        "Processos" => "t_word.gif",
        "Processo" => "t_write.gif"
    );
    var $xmlDepth = array();
    var $treeDepth;
    var $lastElement;
    var $nodeName;
    var $nodeCount ;
    var $maxLevel;

    /******FUNO PARA TAGS DE ABERTURA******/
    function startElement($parser, $name, $attrs) {
        $this->lastElement = $name;
        if($name=="ENTIDADES"){
            $this->node[0]["treeDepth"]="0";
            $this->node[0]["ID"]=trim($attrs["ID"]);
        }

        if(($name=="ENTIDADE")||($name=="ORIGEM")||($name=="FASES")||($name=="FASE")||($name=="PROCESSOS")||($name=="PROCESSO")){
            $this->treeDepth++;
            if($this->treeDepth > $this->maxLevel){
```

```

    $this->maxLevel = $this->treeDepth;
}
$this->nodeCount++;
$this->nodeID = $this->nodeCount;
$this->node[$this->nodeID]["treeDepth"] = $this->treeDepth;
}
while (list($k, $v) = each($attrs)) {
    $this->node[$this->nodeID][$k] = trim($v);
}
$this->xmlDepth[$parser]++;
} // end startElement($parser, $name, $attrs)

/*****FUNCAO TRATAMENTO DOS DADOS ENTRE AS TAGS *****/
function dataElement($parser, $data) {
    $this->node[$this->nodeID][$this->lastElement] .= trim($data);
} // end dataElement($parser, $data)

/*****FUNCAO TRATAMENTO DE TAGS FINAIS*****/
function endElement($parser, $name) {

if(($name=="ENTIDADES")||($name=="ENTIDADE")||($name=="ORIGEM")||($name=="FASES")||($name
=="FASE")||($name=="PROCESSOS")||($name=="PROCESSO")){
    $this->treeDepth--;
    $this->node[$this->nodeID]["last"]=1;
}
$this->xmlDepth[$parser]--;
} // endElement($parser, $name)

/*****FUNCAO Q MONTA A ARVORE*****/
function makeTree($XML_FILE,$e){
    if(get_magic_quotes_runtime()){
        set_magic_quotes_runtime(0);
        $mq = 1;
    }
    $this->parser = xml_parser_create(); //CRIA O PARSER SAX
    xml_set_object($this->parser, &$this);
    //INDICA AS FUNCOES DE MANIPULAO DE TAGS
    xml_set_element_handler($this->parser, "startElement", "endElement");
    //INDICA AS FUNÇÕES DE MANIPULAO DE DADOS DO PARSER
    xml_set_character_data_handler ($this->parser, "dataElement");

    if($XML_FILE){ //verifica se a variavel possui conteudo
        if (!$fp = fopen($XML_FILE, "r")) { //tenta abrir o arquivo
            die("No foi possvel abrir o arquivo");
        }
    }
    else{ //seno tenta abrir o arquivo default na variavel $this->file
        if (!$fp = fopen($this->file, "r")) {
            die("No foi possvel abrir o arquivo");
        }
    }

    while ($data = fread($fp, 4096)) { //l o arquivo xml ate o final
        //aplica funes do parser, se encontrar erro apresenta na tela

```

```

    if (!xml_parse($this->parser, $data, feof($fp)) {
        die(sprintf("XML error: %s at line %d",
            xml_error_string(xml_get_error_code($this->parser)),
            xml_get_current_line_number($this->parser)));
    }
}
xml_parser_free($this->parser);
if($mq){
    set_magic_quotes_runtime (1);
}

return $this->displayTree($e);

} // end makeTree()

//*****FUNO Q MOSTRA A RVORE*****//
function displayTree($e)
{
    // abre o string e num array, separando por vrgula
    $expands = explode(",",$e);
    $totalExpands = count($expands); //conta o total de nodos expandidos
    // marca com 1 no array expands os pontos q devem ser expandidos
    foreach($expands as $x){
        $expand[$x] = 1;
    }
    $visible[0]=1;
    for ($i=0; $i<=$totalExpands; $i++) {
        $aNNode=$expands[$i];
        if ($aNNode<$this->nodeCount && $visible[$aNNode]==1 && $expand[$aNNode]==1) {
            $nextNode=$aNNode+1;
            while ( $this->node[$nextNode]["treeDepth"] > $this->node[$aNNode]["treeDepth"] ) {
                if($this->node[$nextNode]["treeDepth"]== $this->node[$aNNode]["treeDepth"]+1){
                    $visible[$nextNode] = 1;
                    $lastnode = $nextNode;
                }
                $nextNode++;
            }
            $lastNode[$lastnode] = 1;
        }
    }
    $lastlevel = $this->maxLevel;
    // inicia a montagem do html
    $treeHTML = "<table cellspacing=0 cellpadding=0 border=0 cols=".(($this->maxLevel+3)."
width=".(($this->maxLevel*16+$width2)).">\n";
    $treeHTML .= "<tr>";
    for ($i=0; $i<$this->maxLevel; $i++) $treeHTML .= "<td width=20></td>";
    for ($i=0; $i<$this->maxLevel; $i++) $treeLevels[$i]=1;
    for($aNNode=0;$aNNode<$this->nodeCount+1;$aNNode++){ //for 001
        $nextNode = $aNNode+1;
        if($visible[$aNNode]==1){ //if 001
            $treeHTML .= "<tr>\n";
            for ($i=0; $i<$this->node[$aNNode]["treeDepth"]-1; $i++) { //for 002
                if ($treeLevels[$i]==1) {

```

```

        $treeHTML .= "<td><img src=\"\" . $this->imgPath . $this->imgType[\"mbar\"] . \"\"></td>";
    }
    else {
        $treeHTML .= "<td></td>";
    }
} // fim for 002

if ($this->node[$aNodo]["treeDepth"]) { // if 002
    if ($lastNode[$aNodo]) {
        $Img = $this->imgType["mcorner"];
        $treeHTML .= "<td><img src=\"\" . $this->imgPath . $Img . \"\" alt=\"\"></td>";
        $treeLevels[$this->node[$aNodo]["treeDepth"]-1]=0;
    }
    else {
        $Img2 = $this->imgType["mtee"];
        $treeHTML .= "<td><img src=\"\" . $this->imgPath . $Img2 . \"\" alt=\"\"></td>";
        $treeLevels[$this->node[$aNodo]["treeDepth"]-1]=1;
    }
} // fim if 002

if (($this->node[$nextNode]["treeDepth"]>$this->node[$aNodo]["treeDepth"]) { // if 003
    $postString="?e=";
    for ($i=0; $i<$this->nodeCount; $i++) {
        if ($expand[$i]==1 xor $aNodo==$i) {
            if ($postString != "?e=") $postString .= ",";
            $postString .= $i;
        }
    }
    if ($expand[$aNodo]==0) {

        if($lastNode[$aNodo]){
            $treeHTML .= "<td><a href=\"dados.php\" . $postString . "&cod1=" . $_REQUEST['cod1'] .
            "&cod2=" . $_REQUEST['cod2'] . "&cod3=" . $_REQUEST['cod3'] . \"\"><img src=\"\" . $this->imgPath .
            $this->imgType["cplus"] . \"\" border=\"0\" alt=\"expand\"></a></td>\n";
        }
        else{
            $treeHTML .= "<td><a href=\"dados.php\" . $postString . "&cod1=" . $_REQUEST['cod1'] .
            "&cod2=" . $_REQUEST['cod2'] . "&cod3=" . $_REQUEST['cod3'] . \"\"><img src=\"\" . $this->imgPath .
            $this->imgType["tplus"] . \"\" border=\"0\" alt=\"expand\"></a></td>\n";
        }
    }
    else {
        $treeHTML .= "<td><a href=\"dados.php\" . $postString . "&cod1=" . $_REQUEST['cod1'] . "&cod2="
        . $_REQUEST['cod2'] . "&cod3=" . $_REQUEST['cod3'] . \"\"><img src=\"\" . $this->imgPath . $this-
        >imgType["tminus"] . \"\" border=\"0\" alt=\"collapse\"></a></td>\n";
    }
}
else {
    if($this->node[$aNodo]["ID"]=="Origem"){

        $leafImg = $this->imgPath.$this->imgType["Origem"];
    }
}

```

```

else{

    $leafImg = $this->imgPath.$this->imgType["Processo"];

}

if($leafImg){
    $treeHTML .= "<td><img src=\"\" . $leafImg . \"\" alt=\"\"></td>\n";
}
} //fim if 003

$treeHTML .= "<td colspan=" . ($this->maxLevel-$this->node[$aNode]["treeDepth"]+3) . " nowrap >";
$treeHTML .= "<a href=\"\" . \"../telas/dados.php?e=" . $_REQUEST['e'] . "&";

if ($this->node[$aNode]["treeDepth"]==0){
    $treeHTML.="cod1=all&cod2=all&cod3=all";
}
elseif ($this->node[$aNode]["treeDepth"]==1){
    $treeHTML.="cod1=" . $this->node[$aNode]["CODIGO"] . "&cod2=0&cod3=0";
    $cod1 = $this->node[$aNode]["CODIGO"];
}
elseif ($this->node[$aNode]["treeDepth"]==2){
    if($this->node[$aNode]["ID"]!="Origem"){
        $treeHTML.="cod1=" . $cod1 . "&cod2=all&cod3=all";
    }else{
        $treeHTML.="cod1=" . $cod1 . "&cod2=origem&cod3=all";
    }
}
elseif ($this->node[$aNode]["treeDepth"]==3){
    $treeHTML.="cod1=" . $cod1 . "&cod2=" . $this->node[$aNode]["CODIGO"] . "&cod3=0";
    $cod2 = $this->node[$aNode]["CODIGO"];
}
elseif ($this->node[$aNode]["treeDepth"]==4){
    $treeHTML.="cod1=" . $cod1 . "&cod2=" . $cod2 . "&cod3=all";
}
elseif ($this->node[$aNode]["treeDepth"]==5){
    $treeHTML.="cod1=" . $cod1 . "&cod2=" . $cod2 . "&cod3=" . $this->node[$aNode]["CODIGO"];
}

$treeHTML .= "\"";
$treeHTML .= ">";
$treeHTML .= "<font color='#009999' size='2' face='Verdana, Arial, Helvetica, sans-serif'>";
// se for entidade imprime o id e o código senão somente o id
if ($this->node[$aNode]["treeDepth"]==1){
    $treeHTML .= $this->node[$aNode]["ID"] . " " . $this->node[$aNode]["CODIGO"]
."</font></a></td>\n";
}
else {
    $treeHTML .= $this->node[$aNode]["ID"] . "</td>\n";
}

$treeHTML .= "</tr>\n";

```

```

    }//fim if 001
  } //fim for 001
  $treeHTML .= "</table>\n";

  Return $treeHTML;

} // end displayTree()

} // end class
?>

```

Nome do arquivo: dados.php

Descrição: código que mostra os dados do arquivo [entidade].xml. Através da chamada de função fGeraRelatorio() da classe TransformaXML e de acordo com os parâmetros enviados para a mesma, são apresentados os dados de Origem, Processo, Relatório completo, etc.

```

<?
    if(file_exists($caminho)){
        include_once("../classes/clsConverteXML.php");
        $conversor = new clsConverte();
        $conversor-
>fGeraRelatorio($_REQUEST["cod1"],$_REQUEST["cod2"],$_REQUEST["cod3"]);
    }else{
        echo "<font color=#339933 face=Verdana, Arial, Helvetica, sans-serif><b>NÃO hÃ;
nenhum registro da entidade ".$_SESSION['nmEntidade'].</b><br><br><a
href='tela.php?tt=Entidade'>Cadastrar um novo registro</a></font>";
    }

?>

```

Nome do arquivo: tela.php

Descrição: código que chama as funções de geração de telas, da classe TransformaXML. De acordo com os parâmetros que são enviados esta classe gera as telas de cadastro ou edição, de Entidade, Origem, Fase e Processo.

```

<?
    include_once("../classes/clsConverteXML.php");
    if($_REQUEST['tt']!=""){
        if($_REQUEST['tt']=="Entidade"){
            $xml = "../xml/bancoentidades.xml";
            $codTela = $_SESSION['codEntidade'];
            $codEnt = "";
            $codFase = "";
        }
    }

```

```

elseif($_REQUEST['tt']=="Processo"){
    $xml = "../xml/bancoprocessos.xml";
    $codTela = $_REQUEST['cod'];
    $codEnt = $_REQUEST['cod1'];
    $codFase = $_REQUEST['cod2'];
}
elseif($_REQUEST['tt']=="Origem"){
    $xml = "../xml/bancoorigem.xml";
    $codTela = $_REQUEST['cod'];
    $codEnt = $_REQUEST['cod1'];
    $codFase = "";
}
elseif($_REQUEST['tt']=="Fase"){
    $xml = "../xml/bancofases.xml";
    $codTela = $_REQUEST['cod'];
    $codEnt = $_REQUEST['cod1'];
    $codFase = $_REQUEST['cod2'];
}
$conversor = new clsConverte();
$conversor->fGeraTela($codTela,$xml,
$_REQUEST['tt'],$_REQUEST['e'],$codEnt,$codFase);
}
else{
    $xml = "../xml/".$_SESSION['nmEntidade'].".xml";
    $conversor = new clsConverte();
    $conversor-
>fGeraTelaEd($xml,$_REQUEST['cod1'],$_REQUEST['cod2'],$_REQUEST['cod3'],$_REQUEST['e']);
}
?>

```

Nome do arquivo: scriptGavaDados.php

Descrição: este script é chamado quando manda Salvar o registro de uma Entidade, Fase, Origem ou Processo.

```

<?php
include_once('../classes/clsDados.php');

if($_REQUEST['Grava']=="S"){
    $aux2 = 0;
    $control=0;
    $NewVal = "";
    $arrayDados = array();
    while(list($key, $val) = each($_POST)) { //IÃª cada item do array $_POST e coloca o nome da
variÃªvel em $key e o conteudo em $val
        if ($aux2 == 0){
            $tipo = $key;
            $nome = $val;
            $aux2++;
        }
    }
}
else{

```



```

        </script>
    <?
    }elseif($tipo=="Fase"){
    ?>
        <script language="javascript">
            alert("Dados gravados com sucesso!");
            document.location =
            "../dados.php?e=<?=$e?>&cod1=<?=$_REQUEST['codEnt'];?>&cod2=<?=$cod;?>&cod3=0";
        </script>
    <?
    }
    }
?>

```

Nome do arquivo: scriptAlterarDados.php

Descrição: este script é chamado quando manda Alterar o registro de uma Entidade, Fase, Origem ou Processo.

```

<?php
include_once('../classes/clsDados.php');
if($_REQUEST['Grava']=="S"){
    $aux2 = 0;
    $control=0;
    $NewVal = "";
    $arrayDados = array();
    while(list($key, $val) = each($_POST)) { //lÃa cada item do array $_POST e coloca o nome da
variÃvel em $key e o conteudo em $val
        if ($aux2 == 0){
            $tipo = $key;
            $nome = $val;
            $aux2++;
        }
        else{
            $keyTeste = substr($key,0,4 );
            if($keyTeste=="HDN0" || $keyTeste=="HDN1"){
                if ($control==0){
                    $control=1;}
                else{
                    $control=0;
                    $arrayDados[$aux2] = array($val,substr($NewVal,0,strlen($NewVal)-1));
                    $aux2++;
                }
            }
            }else{
                if($control==1){
                    $NewVal .= $val.".";
                }
                else{
                    $arrayDados[$aux2] = array($key,$val);
                    $aux2++;
                }
            }
        }
    }
}

```

```

    }
}
}
array_splice($arrayDados,$aux2-2,1); //remove o último item do array que Ã(c) a referÃancia do
botÃ£o salvar.
$entidade = new clsDados("../xml/".$_SESSION['nmEntidade'].".xml");
$entidade->setNome($nome);
$entidade->setTipo($tipo);
$entidade->setCodEnt($_REQUEST['cod1']);
$entidade->setCodFase($_REQUEST['cod2']);
$entidade->setCodProc($_REQUEST['cod3']);
$e = $_REQUEST['e'];
$cod = $entidade->fAlterar($arrayDados);
if ($cod != ""){
    if($tipo=="Entidade"){
        ?>
        <script language="javascript">
            alert("Dados alterados com sucesso!");
            document.location =
"./dados.php?e=<?=$e?>&cod1=<?=$cod;?>&cod2=0&cod3=0";
        </script>
        <?
    }elseif($tipo=="Processo"){
        ?>
        <script language="javascript">
            alert("Dados alterados com sucesso!");
            document.location =
"./dados.php?e=<?=$e?>&cod1=<?=$_REQUEST['cod1'];?>&cod2=<?=$_REQUEST['cod2'];?>&cod3=<
?=$cod;?>";
        </script>
        <?
    }elseif($tipo=="Origem"){
        ?>
        <script language="javascript">
            alert("Dados alterados com sucesso!");
            document.location =
"./dados.php?e=<?=$e?>&cod1=<?=$_REQUEST['cod1'];?>&cod2=origem&cod3=all";
        </script>
        <?
    }elseif($tipo=="Fase"){
        ?>
        <script language="javascript">
            alert("Dados alterados com sucesso!");
            document.location =
"./dados.php?e=<?=$e?>&cod1=<?=$_REQUEST['cod1'];?>&cod2=<?=$cod;?>&cod3=0";
        </script>
        <?
    }
}
}
?>

```

Nome do arquivo: scriptExcluiDados.php

Descrição: este script é chamado pelo link Excluir, presente nos registros de Entidade e de Processo.

```
<?php
    $res = "";
    include_once('../classes/clsDados.php');
    $entidade = new clsDados("../xml/".$_SESSION['nmEntidade'].".xml");
    $entidade->setTipo($_REQUEST['tp']);
    $entidade->setCodEnt($_REQUEST['cod1']);
    $entidade->setCodFase($_REQUEST['cod2']);
    $entidade->setCodProc($_REQUEST['cod3']);
    $e = $_REQUEST['e'];
    if ($entidade->fExclui()){
        if($_REQUEST['tp']=="Entidade"){
            ?>
                <script language="javascript">
                    alert("Dados apagados com sucesso!");
                    document.location = "../dados.php?e=<?=$e?>&cod1=all&cod2=all&cod3=all";
                </script>
            <?
        }elseif($_REQUEST['tp']=="Processo"){
            ?>
                <script language="javascript">
                    alert("Dados apagados com sucesso!");
                    document.location =
"../dados.php?e=<?=$e?>&cod1=<?=$_REQUEST['cod1'];?>&cod2=<?=$_REQUEST['cod2'];?>&cod3=0";
                </script>
            <?
        }
    }
?>
```

Nome do arquivo: agrupa.php

Descrição: arquivo que monta a tela de agrupamento de entidades. Quando o operador clica no botão Agrupar, chama a função agrupar() da classe dados.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<?php
    include("login.inc");
    include_once('../classes/clsDados.php');
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script language="JavaScript">

    function enviaDados(){
        vetCodigo = new Array();
        var x = 0;
        for(x;x<document.getElementById("Agrupar").length;x++){
            vetCodigo.splice(x,0,document.getElementById("Agrupar").options[x].value);
        }
        document.getElementById('hdnCodigos').value=vetCodigo;
    }

    function Inclui(){
        var text =
document.getElementById("Entidades").options[document.getElementById("Entidades").selectedIndex].te
xt;
        var value =
document.getElementById("Entidades").options[document.getElementById("Entidades").selectedIndex].va
lue;
        document.getElementById("Agrupar").options[document.getElementById("Agrupar").length++] =
new Option(text, value, false, false);

document.getElementById("Entidades").options[document.getElementById("Entidades").selectedIndex] =
null;
    }
    function Exclui(){
        var text =
document.getElementById("Agrupar").options[document.getElementById("Agrupar").selectedIndex].text;
        var value =
document.getElementById("Agrupar").options[document.getElementById("Agrupar").selectedIndex].value;

document.getElementById("Entidades").options[document.getElementById("Entidades").length++] = new
Option(text, value, false, false);

document.getElementById("Agrupar").options[document.getElementById("Agrupar").selectedIndex] = null;
    }
    window.defaultStatus=".: Sistema de Rastreabilidade"
</script>
</title>
<?>
```

```

$entidade = new clsDados("../xml/".$_SESSION['nmEntidade'].".xml");
if($_REQUEST['gravar']=="S"){
    $entidade->setCodEnt($_REQUEST['cod1']);
    if($entidade->agrupa($_POST['hdnCodigos'])){
        ?>
        <script language="javascript">
            alert("Agrupamento gravado com sucesso!");
            document.location =
"dados.php?e=<?=$_REQUEST['e']?>&cod1=<?=$_REQUEST['cod1']?>&cod2=0&cod3=0";
        </script>
        <?
    }
}
?>
<title>Sistema rastreabilidade</title>
</head>
<body>
<div align="center">
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="227" height="157" valign="top"><p align="left"></p>
<p align="left"></p></td>
<td width="727" valign="top"><table width="100%" border="0">
<tr>
<td width="89%" valign="top"><div align="right"><strong><font color="#009999" size="2"
face="Verdana, Arial, Helvetica, sans-serif">
<? echo "usuário:&nbsp;".$_SESSION['lognome']; ?>
</font></strong></div></td>
<td width="11%" valign="top"><div align="right"><strong><a href="logout.php"
target="_parent"><font color="#009999" size="2" face="Verdana, Arial, Helvetica, sans-
serif">Sair</font></a></strong></div></td>
</tr>
<tr>
<td height="125" align="left" valign="middle"><p><font color="#339966" face="Arial, Helvetica,
sans-serif">
<?
                echo "Objeto da rastreabilidade:&nbsp;&nbsp;&nbsp;<font
size=5>".$_SESSION['nmEntidade']."</font>";
                ?>
            </font></p></td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
</table></td>
</tr>
<tr>
<td height="204" valign="top">
<table width="100%">
<tr>
<td height="41" valign="top"><span class="style3"><strong><a
href="tela.php?tt=Entidade&e=<?=$_REQUEST['e'];?>&cod1=<?=$_REQUEST['cod1'];?>&cod2=<?=$_R
EQUEST['cod2'];?>&cod3=<?=$_REQUEST['cod3'];?>"><font color="#009900" face="Verdana, Arial,
Helvetica, sans-serif">NOVO(A) REGISTRO </font></a></strong></span></td>

```

```

</tr>
<tr>
  <td height="19"><p><span class="style3"><strong><font color="#009900" face="Verdana, Arial,
Helvetica, sans-serif">CADASTRADOS</font></strong></span></p></td>
</tr>
<tr>
  <td height="29">
    <?
      include("arvorexml/xmltree.php");
      $tree = new tree;
      $caminho = "../xml/" . $_SESSION['nmEntidade'] . ".xml";
      if(file_exists($caminho)){
        $html = $tree->makeTree($caminho,$_REQUEST["e"]);
        print $html . "<br>";
      }else{
        echo "No h registros";
      }
    ?>
  </td>
</tr>
<tr>
  <td height="24">&nbsp;</td>
</tr>
<tr>
  <td height="24"><a href="altera_senha.php"><font color="#009999" size="2" face="Verdana, Arial,
Helvetica, sans-serif">Alterar
  minha senha</font></a></td>
<tr>
  <td height="24"><a href="principal_user.php"><font color="#009999" size="2" face="Verdana,
Arial, Helvetica, sans-serif">Outro
  tipo de entidade</font></a></td>

</table>
  <td valign="top"><form name="dados" method="post"
action="agrupar.php?e=<?=$_REQUEST['e']?>&gravar=S&cod1=<? echo $_REQUEST['cod1']?>"
onSubmit="enviaDados()">
  <input type="hidden" name="hdnCodigos">
  <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0" id="tbl">
    <tr>
      <td colspan="3">&nbsp;</td>
    </tr>
    <tr bgcolor="#339933">
      <th width="8%" height="19" bgcolor="#339933"><div align="left"><font color="#FFFFFF" size="3"
face="Verdana, Arial, Helvetica, sans-serif">Agrupar</font></div></th>
      <th colspan="2"><div align="left"><font color="#FFFFFF" size="3" face="Verdana, Arial, Helvetica,
sans-serif">&nbsp;<? echo $_SESSION['nmEntidade']?>&nbsp;<? echo
$_REQUEST['cod1']?></font></div></th>
    </tr>
    <tr>
      <td height="36" colspan="3" id="cel"></td>
    </tr>
    <tr>
      <td id="cel">
        <p><font size="2" face="Verdana, Arial, Helvetica, sans-serif">Entidades</font>:</p>

```



```

function enviaDados(){
    vetCodigo = new Array();
    var x = 0;
    for(x;x<document.getElementById("Agrupar").length;x++){
        vetCodigo.splice(x,0,document.getElementById("Agrupar").options[x].value);
    }
    document.getElementById('hdnCodigos').value=vetCodigo;
}

function Inclui(){
    var text =
document.getElementById("Entidades").options[document.getElementById("Entidades").selectedIndex].text;
    var value =
document.getElementById("Entidades").options[document.getElementById("Entidades").selectedIndex].value;
    document.getElementById("Agrupar").options[document.getElementById("Agrupar").length++] = new
Option(text, value, false, false);
    document.getElementById("Entidades").options[document.getElementById("Entidades").selectedIndex] =
null;
}

function Exclui(){
    var text =
document.getElementById("Agrupar").options[document.getElementById("Agrupar").selectedIndex].text;
    var value =
document.getElementById("Agrupar").options[document.getElementById("Agrupar").selectedIndex].value;
    document.getElementById("Entidades").options[document.getElementById("Entidades").length++] = new
Option(text, value, false, false);
    document.getElementById("Agrupar").options[document.getElementById("Agrupar").selectedIndex] =
null;
}

window.defaultStatus=".: Sistema de Rastreabilidade"
</script>
</title>
<?
$entidade = new clsDados("../xml/XMLDados/" .$_SESSION['nmEntidade'].".xml");
if($_REQUEST['gravar']=="S"){
    $entidade->setCodEnt($_POST['hdnCodigos']);
    if($entidade->desagrupa($_REQUEST['cod1'])){
        ?>
        <script language="javascript">
            alert("Desagrupamento gravado com sucesso!")
            document.location =
"dados.php?e=<?=$_REQUEST['e']?>&cod1=<?=$_REQUEST['cod1']?>&cod2=0&cod3=0";
        </script>
        <?
    }
}

?>
</head>
<body>
<div align="center">
</table> </td>

```


CLASSES

Classe: XML

Descrição: função que faz o parser dos arquivos XML. É a única classe que abre e que grava arquivos XML. Também possui algumas funções de inclusão, alteração e exclusão de dados.

```
<?php
class clsXML
{
    var $root, $doc , $tag, $arquivo, $subtag, $tagcod, $control;
    function fGravar($arraydados,$nmTag)
    {
        $cod = $this->fCodigo($this->tagcod);
        $this->fTag($nmTag,"/");
        $this->fAtributo('CODIGO',$cod);
        $this->fAtributo('ATIVO',"S");
        while (list($key1) = each($arraydados))
        {
            $tagnome = $arraydados[$key1][1];
            $conteudo = $arraydados[$key1][0];

            if($tagnome == "valores" && $nmTag == "campo"){
                $this->fTag("valores","/");
                $opcoes = explode(";", $conteudo);
                foreach($opcoes as $opcao){
                    $this->fTag("opcao", $opcao);
                }
            }
            else
            {
                $this->fTag($tagnome,$conteudo);
            }

        }
    }
    //fim while
    if ($this->fSalvar()) //salva o arquivos
    {
        return $cod;
    }else return "0";
} //fim função

function fEditar($arraydados,$codcampo,$nmTag)
{
    $this->fTag($nmTag,"/");
    $this->fAtributo('CODIGO',$codcampo);
    $this->fAtributo('ATIVO',"S");

    while (list($key1) = each($arraydados))
    {
        $tagnome = $arraydados[$key1][1];
        $conteudo = $arraydados[$key1][0];
    }
}
```

```

        if($tagname == "valores" && $nmTag == "campo"){
            $this->fTag("valores","/");
            $opcoes = explode(";", $conteudo);
            foreach($opcoes as $opcao){
                $this->fTag("opcao", $opcao);
            }
        }
        else
        {
            $this->fTag($tagname, $conteudo);
        }
    }

} //fim while

$items = $this->root->getElementsByTagName($nmTag);
foreach ($items as $item) {
    if ($item->getAttribute('CODIGO') == $codcampo){
        $childremove = $this->root->replaceChild($this->tag, $item);
    }
}
if ($this->fSalvar()) //salva o arquivos
{
    return true;
}
}

function fExcluir($campo, $tag)
{
    $this->root = $this->doc->documentElement;
    $items = $this->root->getElementsByTagName($tag);
    foreach ($items as $item) {
        if ($item->getAttribute("CODIGO") == $campo){
            $item->setAttribute("ATIVO", "N");
        }
    }
    if ($this->fSalvar()) //salva o arquivos
    {
        return true;
    }
}

function fArquivo($arquivo, $root)
{
    $this->arquivo = $arquivo;
    if (file_exists($this->arquivo)) {
        $this->doc = DOMDocument::load($this->arquivo);
        $this->tagcod = $root;
        $this->root = $this->doc->documentElement; //coloca em root a referencia ao primeiro
nodo do arquivo
        $this->rootcopy = $this->root; //copia root para rootcopy para não criar problemas com a
função campo alterar q precisa de root
    }
    else

```

```

    {
        $this->doc = new DOMDocument('1.0','ISO-8859-1');
        $this->tagcod = $root;
        $this->root = $this->doc->createElement($root);
        $this->root = $this->doc->appendChild($this->root);
        $this->rootcopy = $this->root; //copia root para rootcopy para não criar problemas com a
função campo alterar q precisa de root
    }
}

//inclui o atributo na tag pai
function fAtributo($nome,$valor)
{
    $this->tag->setAttribute($nome,$valor);
}

// inclui atributo nas tags filhas
function fAtributoF($nome,$valor)
{
    $this->subtag->setAttribute($nome,$valor);
}

function fTag($tagnome,$conteudo)
{
    if ($conteudo == "//")
    {
        $this->tag = $this->doc->createElement($tagnome);
        $this->rootcopy = $this->rootcopy->appendChild($this->tag);
        $this->tag = $this->rootcopy; //faz uma copia de rootcopy para tag para q se na proxima
vez ocorrer else ele tenha a referencia correta do nodo
        $this->control++;
        if($this->control == 1){$this->rootcopy2 = $this->rootcopy;}
    }
    else
    {
        $this->subtag = $this->doc->createElement($tagnome,$conteudo);
        $this->subtag = $this->tag->appendChild($this->subtag);
    }
}

/*****Função que gera o código do próximo item a ser incluso*****/
function fCodigo($nometag)
{
    $items = $this->doc->getElementsByTagName($nometag);
    foreach ($items as $item) {
        $tag = $item->lastChild;
        if ($tag){
            $cod = $tag->getAttribute('CODIGO');
            $cont=$cod+1;
            return($cont);
        }
        else{
            return(1);
        }
    }
}

```

```

    }
}

/*****Função responsável por salvar os arquivos xml *****/
function fSalvar()
{
    $this->doc->save($this->arquivo);
    return true;
}
} //fim classe xml
?>

```

Classe: Campos

```

<?
//CLASSE: clsCampos - Classe para criação, alteração e exclusão de campos.
include_once('clsXML.php');

class clsCampos
{
    private $codigo, $nome, $thtml, $defaultt, $arquivoXML;

    // classe construtora da classe
    public function clsCampos($codcampo,$caminho)
    {
        $this->arquivoXML = $caminho;
        if(file_exists($caminho)){
            $xml = simplexml_load_file($caminho);
            foreach($xml as $doc)
            {
                if ($doc['CODIGO']==$codcampo)
                {
                    $this->setNome($doc->nome);
                    $this->setThtml($doc->thtml);
                    $valores="";
                    foreach($doc->valores->opcao as $it)
                    {
                        $valores .= $it.";";
                    }

                    $this->setDefault(substr($valores,0,strlen($valores)-1));

                } //fim if
            } // fim foreach
        }else{
            $this->setNome("");
            $this->setThtml("");
            $this->setDefault("");
        }
    } // fim função getcampo
}

```

```

public function fMontaArray()
{
    $arraycampos = array(array("$this->nome", "nome"), array("$this->tphtml", "tphtml"), array("$this-
>defaultt", "valores"));
    return($arraycampos);
}

public function incluir()
{
    $xml = new clsXml();
    $arrayCampo = $this->fMontaArray();
    $xml->fArquivo($this->arquivoXML, 'banco_campos');
    $cod = $xml->fGravar($arrayCampo, 'campo');
    return $cod;
}

public function excluir($codcampo)
{
    $xml = new clsXml();
    $xml->fArquivo($this->arquivoXML, 'banco_campos');
    if ($xml->fExcluir($codcampo, 'campo'))
    {
        return true;
    }
}

public function alterar($codcampo)
{
    $xml = new clsXml();
    $arrayCampo = $this->fMontaArray();
    $xml->fArquivo($this->arquivoXML, 'banco_campos');
    if ($xml->fEditar($arrayCampo, $codcampo, 'campo'))
    {
        return true;
    }
}

//função que cria um array com os campos cadastrados no sistema
function fMontaLista()
{
    $xml = simplexml_load_file($this->arquivoXML);
    $aux="";
    $Resultado = "";
    $ArrayResultado = array(array(), array());
    $x="0";
    foreach ($xml as $doc)
    {
        if($doc['ATIVO']=="S"){
            $valores="";
            $aux .= $doc['CODIGO']."|";
            $aux .= $doc->nome."|";
            $aux .= $doc->tphtml."|";
            foreach($doc->valores->opcao as $it)
            {

```

```

                $valores .= $it."";
            }
            $valores = substr($valores,0,strlen($valores)-1);
            $aux .= $valores."|";
            $ArrayResultado[$x][1] = $aux;
            $ArrayResultado[$x][0] = $doc->nome;
            $x++;
            $aux = "";
            $valores="";
        }
    } // fim foreach
    foreach($ArrayResultado as $res)
        $sortAux[] = $res[0];
    array_multisort($sortAux, SORT_ASC, $ArrayResultado);
    for($y=0;$y<$x;$y++){
        $Resultado .= "<option
        value=".$ArrayResultado[$y][1].">".$ArrayResultado[$y][0]."</option>";
    }
    return $Resultado;
}

} //fim classe

?>

```

Classe: Usuário

```

<?
//CLASSE: clsUsuarios - Classe para criação, alteração e exclusão de Usuarios.
include_once('clsXML.php');
class clsUsuarios
{
    private $codigo, $nome, $Tipo, $Login, $senha;
    // construtor da classe
    public function clsUsuarios($codigo,$xml)
    {
        $xml = simplexml_load_file($xml);
        foreach($xml as $doc)
        {
            if ($doc['CODIGO']==$codigo)
            {
                $this->setNome($doc->nome);
                $this->setTipo($doc->tipo);
                $this->setLogin($doc->login);
                $this->setSenha($doc->senha);
            } //fim if
        } // fim foreach
    } // fim função setuSUARIOS

    public function fMontaArray()
    {
        $arrayUsuarios = array(array("$this->nome", "nome"), array("$this->Tipo", "tipo"), array("$this->Login", "login"), array("$this->senha", "senha"));
        return($arrayUsuarios);
    }
}

```

```

    }

    public function incluir()
    {
        $xml = new clsXml();
        $arrayCampo = $this->fMontaArray();
        $xml->fArquivo('../xml/usuarios.xml','usuarios');
        if ($xml->fGravar($arrayCampo,'usuario')!="0")
        {
            return true;
        }
    }

    public function excluir($codcampo)
    {
        $xml = new clsXml();
        $xml->fArquivo('../xml/usuarios.xml','usuarios');
        if ($xml->fExcluir($codcampo,'usuario'))
        {
            return true;
        }
    }

    public function alterar($codcampo)
    {
        $xml = new clsXml();
        $arrayCampo = $this->fMontaArray();
        $xml->fArquivo('../xml/usuarios.xml','usuarios');
        if ($xml->fEditar($arrayCampo,$codcampo,'usuario'))
        {
            return true;
        }
    }

} //fim classe

?>

```

Classe: Acervo (Tela)

```

<?
    include_once('clsXML.php');
class clsTela
{
    private $campos, $Nome, $Tipo, $Origem, $Metadado, $xml;

    public function setTela($codigo)
    {
        $xml = simplexml_load_file($this->xml);
        $campotela = "";
        $metadados = "";
        foreach($xml as $doc)
        {

```

```

if ($doc['CODIGO']==$codigo)
{
    $this->setNome($doc['nome']);
    $this->setTipo($doc['tipo']);
    foreach($doc->campo as $campo){
        $campotela .= $campo['CODIGO'].", ";
    }
    $this->setCampos($campotela);
    if($this->Tipo=="Entidade"){
        $this->setOrigem($doc->Origem);
        foreach($doc->Fase as $Fase){
            $metadados .= $Fase.", ";
        }
        $this->setMetadado(substr($metadados,0,strlen($metadados)-1));
    }
    if($this->Tipo=="Fase"){
        foreach($doc->Processo as $Processo){
            $metadados .= $Processo.", ";
        }
        $this->setMetadado(substr($metadados,0,strlen($metadados)-1));
    }
} //fim if
} // fim foreach
} // fim função setTela

```

/******ESTA FUNÇÃO GRAVA EM UM ARQUIVO XML OS DADOS DA TELA *****/

```

function fSalvaTela()
{
    $campos = new clsXml();
    $tela = new clsXml();

    $campos->fArquivo('./xml/XMLEstrutura/bancocampo.xml','banco_campos');
    $tela->fArquivo($this->xml,'banco_telas');

    $cd = $tela->fCodigo($tela->tagcod);// esta linha tem que vir antes da linha ftag
    $tela->fTag(strtoupper($this->Tipo),"/");
    $tela->fAtributo('CODIGO',$cd);
    $tela->fAtributo('ATIVO',"S");
    $tela->fAtributo('tipo',$this->Tipo);
    $tela->fAtributo('nome',$this->Nome);

    $codCampos = explode(",",$this->campos);
    $cont = count($codCampos);

    for ($x=0;$x<=$cont;$x++){
        //faz a leitura do arquivo bancocampos.xml para encontrar o campo incluído na tela
        $items = $campos->doc->getElementsByTagName('campo');
        foreach ($items as $item) {
            if ($item->getAttribute('CODIGO') == $codCampos[$x]){
                $arq = $tela->doc->documentElement;
                $tagcampo = new clsXml();
                $tagcampo->tag = $tela->doc->importNode($item, true);
                $arq = $tela->rootcopy2->appendChild($tagcampo->tag);
            }
        }
    }
}

```

```

    }
}

if($this->Tipo=="Entidade"){
    $tela->fTag("Origem",$this->Origem);
    $codMetadado = explode(",",$this->Metadado);
    $cont = count($codMetadado);
    for ($x=0;$x<$cont;$x++){
        $tela->fTag("Fase",$codMetadado[$x]);
    }
}elseif($this->Tipo=="Fase"){
    $codMetadado = explode(",",$this->Metadado);
    $cont = count($codMetadado);
    for ($x=0;$x<$cont;$x++){
        $tela->fTag("Processo",$codMetadado[$x]);
    }
}

if ($tela->fSalvar())
{
    return true;
}
}

public function editar($codTela)
{
    $campos = new clsXml();
    $tela = new clsXml();
    $campos->fArquivo('../xml/XMLEstrutura/bancocampo.xml','banco_campos');
    $tela->fArquivo($this->xml,'banco_telas');
    $tela->fTag(strtoupper($this->Tipo),'//');
    $tela->fAtributo('CODIGO',$codTela);
    $tela->fAtributo('ATIVO','S');
    $tela->fAtributo('tipo',$this->Tipo);
    $tela->fAtributo('nome',$this->Nome);
    $codCampos = explode(",",$this->campos);
    $cont = count($codCampos);

    for ($x=0;$x<=$cont;$x++){
        //faz a leitura do arquivo bancocampos.xml para encontrar o campo incluido na tela
        $items = $campos->doc->getElementsByTagName('campo');
        foreach ($items as $item) {
            if ($item->getAttribute('CODIGO') == $codCampos[$x]){
                $arq = $tela->doc->documentElement;
                $tagcampo = new clsXml();
                $tagcampo->tag = $tela->doc->importNode($item, true);
                $arq = $tela->rootcopy2->appendChild($tagcampo->tag);
            }
        }
    }
}

if($this->Tipo=="Entidade"){
    $tela->fTag("Origem",$this->Origem);
}

```



```

function fMontaCombo($arquivo,$tag,$tagname)
{
    $lista = new clsXML();
    $ArrayResultado = array(array(),array());
    $Resultado="";
    $lista->fArquivo($arquivo,$tag);
    $items = $lista->doc->getElementsByTagName($tagname);
    $x="0";
    foreach ($items as $item) {
        if($item->getAttribute('ATIVO')==="S"){
            $ArrayResultado[$x][0] = $item->getAttribute('CODIGO');
            $ArrayResultado[$x][1] = $item->getAttribute('nome');
            $x++;
        }
    }
    foreach($ArrayResultado as $res)
        $sortAux[] = $res[1];
    array_multisort($sortAux, SORT_ASC, $ArrayResultado);
    for($y=0;$y<$x;$y++){
        $Resultado .= "<option
value=" . $ArrayResultado[$y][0]. ">" . $ArrayResultado[$y][1]. "</option>";
    }
    return $Resultado;
}
} // fim classe
?>

```

Classe: Dados

Descrição: esta classe gerencia a gravação de registros de Entidade, Fase, Processo e Origem, nos arquivos de dados.

```

<?
include_once('clsXML.php');
include_once('clsCampos.php');
include_once('clsTela.php');

class clsDados
{
    private $child, $xml, $codEnt, $codFase, $codProc, $nome, $tipo;
    /***** Função construtora da classe *****/
    function clsDados($arquivo)
    {
        $this->xml = new clsXml();
        $this->xml->fArquivo($arquivo,'ENTIDADES');
        $this->xml->root->setAttribute('ID',"Entidades");

        $this->child = new clsXml();
        $this->child->fArquivo('child.xml','item');
    }

    function fGrava($arrayDados,$codTela)
    {

```

```

if (strtoupper($this->tipo) == "ORIGEM"){
    $this->fOrigem($arrayDados);
    $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."/]ORIGEM";
    $busca1 = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."]";
    if($this->fSobrescreve($busca,$busca1)){
        return "0";
    }
}
}elseif (strtoupper($this->tipo) == "ENTIDADE"){
    $cod = $this->fEntidade($arrayDados,$codTela);
    $busca = "/ENTIDADES";
    if($this->fInclui($busca)){
        return $cod;
    }
}
}elseif (strtoupper($this->tipo) == "FASE"){
    $this->fFase($arrayDados,$codTela);
    $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."/]
FASES/FASE[@CODIGO=".$this->codFase."]";
    $busca1 = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."/]FASES";
    if($this->fSobrescreve($busca,$busca1)){
        return "0";
    }
}
}elseif (strtoupper($this->tipo) == "PROCESSO"){
    $cod = $this->fProcesso($arrayDados);
    $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this-
>codEnt."/]FASES/FASE[@CODIGO=".$this->codFase."/]PROCESSOS";
    if($this->fInclui($busca)){
        return $cod;
    }
}
}
}

function fAlterar($arrayDados)
{
    $this->child->fTag(strtoupper("DADOS"),"/");
    $this->fCampos($arrayDados);
    if (strtoupper($this->tipo) == "ORIGEM"){
        $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."/]ORIGEM/CAMPO";
        if($this->fSubstituiCampo($busca)){
            return "0";
        }
    }
    elseif (strtoupper($this->tipo) == "ENTIDADE"){
        $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."/]CAMPO";
        if($this->fSubstituiCampo($busca)){
            return $this->codEnt;
        }
    }
    elseif (strtoupper($this->tipo) == "FASE"){
        $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this-
>codEnt."/]FASES/FASE[@CODIGO=".$this->codFase."/]CAMPO";
        if($this->fSubstituiCampo($busca)){
            return $this->codFase;
        }
    }
    elseif (strtoupper($this->tipo) == "PROCESSO"){
        $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this-
>codEnt."/]FASES/FASE[@CODIGO=".$this-
>codFase."/]PROCESSOS/PROCESSO[@CODIGO=".$this->codProc."/]CAMPO";
    }
}

```

```

        if($this->fSubstituiCampo($busca)){
            return $this->codProc;
        }
    }
}

function fExclui()
{
    if (strtoupper($this->tipo) == "ENTIDADE"){
        $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."]";
        $busca1 = "/ENTIDADES";
        if($this->fRemove($busca,$busca1)){
            return true;
        }
    }elseif (strtoupper($this->tipo) == "PROCESSO"){
        $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."]/FASES/FASE[@CODIGO=".$this->codFase."]/PROCESSOS/PROCESSO[@CODIGO=".$this->codProc."]";
        $busca1 = "/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."]/FASES/FASE[@CODIGO=".$this->codFase."]/PROCESSOS";
        if($this->fRemove($busca,$busca1)){
            return true;
        }
    }
}

private function fProcesso($arrayDados)
{
    $this->child->fTag(strtoupper($this->tipo),"/");
    $this->child->fAtributo('ID',$this->nome);
    $cod = $this->getCodigo("/ENTIDADES/ENTIDADE[@CODIGO=".$this->codEnt."]/FASES/FASE[@CODIGO=".$this->codFase."]/PROCESSOS");
    $this->child->fAtributo('CODIGO',$cod);
    $this->fCampos($arrayDados);
    return $cod;
}

private function fFase($arrayDados,$codTela)
{
    $this->child->fTag(strtoupper($this->tipo),"/");
    $this->child->fAtributo('CODIGO',$this->codFase);
    $this->child->fAtributo('ATIVO','S');
    $this->child->fAtributo('tipo','Fase');
    $this->child->fAtributo('ID',$this->nome);
    $import = new clsTela();
    $import->setXML('../xml/XML Estrutura/bancofases.xml');
    $import->setTela($codTela);
    $processos = $import->GetMetadado();
    $this->fCampos($arrayDados);
    $this->child->fTag('PROCESSOS','/');
    $this->child->fAtributo('ID',"Processos");
    $processos = explode(" ",$processos);
    foreach($processos as $codigoProc){
        $import = new clsTela();
    }
}

```

```

$import->setXML('../xml/XMLEstrutura/bancoprocessos.xml');
$import->setTela($codigoProc);

$this->child->fTag('PROC_ADD', "");
$this->child->fAtributoF('CODIGO', $codigoProc);
$this->child->fAtributoF('tipo', "Processo");
$this->child->fAtributoF('ATIVO', "N");
$this->child->fAtributoF('NOME', $import->GetNome());
}
}

private function fOrigem($arrayDados){
    $this->child->fTag(strtoupper($this->tipo), "");
    $this->child->fAtributo('ATIVO', "S");
    $this->child->fAtributo('nome', $this->nome);
    $this->child->fAtributo('ID', $this->tipo);
    if($arrayDados!=""){ $this->fCampos($arrayDados); }
}

private function fEntidade($arrayDados, $codTela){
    $this->child->fTag(strtoupper($this->tipo), "");
    $this->child->fAtributo('ID', $this->nome);
    $cod = $this->getCodigo("/ENTIDADES");
    $this->child->fAtributo('CODIGO', $cod);
    $this->child->fAtributo('ATIVO', "S");
    $this->fCampos($arrayDados);

    $import = new clsTela();
    $import->setXML('../xml/XMLEstrutura/bancoentidades.xml');
    $import->setTela($codTela);
    $origem = $import->GetOrigem();
    $fases = $import->GetMetadado();
    $import = new clsTela();
    $import->setXML('../xml/XMLEstrutura/bancoorigem.xml');
    $import->setTela($origem);

    $this->child->fTag('ORIGEM', "");
    $this->child->fAtributoF('CODIGO', $origem);
    $this->child->fAtributoF('ATIVO', "N");
    $this->child->fAtributoF('nome', $import->GetNome());
    $this->child->fAtributoF('ID', "Origem");

    $this->child->fTag('FASES', "");
    $this->child->fAtributo('ID', "Fases");

    $fases = explode(", ", $fases);
    $codFase = "1";
    foreach($fases as $codigoTela){

        $import = new clsTela();
        $import->setXML('../xml/XMLEstrutura/bancofases.xml');
        $import->setTela($codigoTela);

        $this->child->fTag('FASE', "");
    }
}

```

```

        $this->child->fAtributoF('CODIGO',$codFase);
        $this->child->fAtributoF('CODIGO_TELA',$codigoTela);
        $this->child->fAtributoF('ATIVO',"N");
        $this->child->fAtributoF('tipo',"Fase");
        $this->child->fAtributoF('ID',$import->GetNome());
        $codFase++;
    }
    return $cod;
}

private function fCampos($arrayDados)
{
    while (list($key) = each($arrayDados))
    {
        $campo = new clsCampos($arrayDados[$key][0], '../xml/XMLEstrutura/bancocampo.xml');
        if($campo->GetTphtml()=="checkbox"){
            $valoresSelecioandos = explode(";", $arrayDados[$key][1]);
            $valores = explode(";", $campo->GetDefault());
            foreach($valores as $val1){
                foreach($valoresSelecioandos as $val2){
                    if(trim($val1)==trim($val2)){
                        $res = "1";
                    }
                }
            }
            if($res=="1"){
                $ValorFinal .= "1;";
            }else{
                $ValorFinal .= "0;";
            }
            $res = "0";
        }
        $this->child->fTag('CAMPO',$ValorFinal);
    }else{
        $this->child->fTag('CAMPO',$arrayDados[$key][1]);
    }

    $this->child->fAtributoF('CODIGOCAMPO',$arrayDados[$key][0]);
    $this->child->fAtributoF('NOME',$campo->GetNome());
    $this->child->fAtributoF('TPHTML',$campo->GetTphtml());
    $this->child->fAtributoF('VALORES',$campo->GetDefault().";");
} // fim while
}

private function fSobrescreve($busca,$busca1)
{
    /**captura no arquivo montado pela função flIncluiChild a tag para ser importada**/
    $items = $this->child->doc->getElementsByTagName(strtoupper($this->tipo));

    foreach ($items as $item) {

        $xpath = new DOMXPath($this->xml->doc); //a variável xpath recebe um objeto
        DOMXPath
        $tag = $this->xml->doc->importNode($item, true); // a variável tag importa o nodo
        montado na função flIncluiChild
    }
}

```

```

        $results = $xpath->query($busca);
        $path = $xpath->query($busca1);
        $path = $path->item(0);
        foreach ($results as $result) {
            $arq = $path->replaceChild($tag,$result);
        }
    }
    if($this->xml->fSalvar()){
        return true;
    }
}

private function fRemove($busca,$busca1)
{
    $xpath = new DOMXPath($this->xml->doc); //a variável xpath recebe um objeto
DOMXPath
    $results = $xpath->query($busca);
    $path = $xpath->query($busca1);
    $path = $path->item(0);
    foreach ($results as $result) {
        $childremove = $path->removeChild($result);
    }
    if($this->xml->fSalvar()){
        return true;
    }
}

private function fInclui($busca)
{
    /**captura no arquivo montado pela função fIncluiChild a tag para ser importada**/
    $items = $this->child->doc->getElementsByTagName(strtoupper($this->tipo));

    foreach ($items as $item) {
        $xpath = new DOMXPath($this->xml->doc); //a variável xpath recebe um objeto
DOMXPath
        $tag = $this->xml->doc->importNode($item, true); // a variável tag importa o nodo
montado na função fIncluiChild
        $path = $xpath->query($busca);
        foreach ($path as $xpath) {
            $caminho = $xpath->appendChild($tag);
        }
    }
    if($this->xml->fSalvar()){
        return true;
    }
}

private function fSubstituiCampo($busca)
{
    $items = $this->child->doc->getElementsByTagName("CAMPO");
    foreach ($items as $item) {
        $xpath = new DOMXPath($this->xml->doc); //a variável xpath recebe um objeto DOMXPath
        $tag = $this->xml->doc->importNode($item, true); // a variável tag importa o nodo
        $results = $xpath->query($busca);
    }
}

```

```

        foreach ($results as $result) {
            if($result->getAttribute("CODIGOCAMPO")==$item->getAttribute("CODIGOCAMPO"))
                $result->nodeValue = $item->nodeValue;
        }
    }
    if($this->xml->fSalvar()){
        return true;
    }
}

private function getCodigo($caminho){
    $xpath = new DOMXPath($this->xml->doc); //a variável xpath recebe um objeto DOMXPath
    $path = $xpath->query($caminho);
    foreach ($path as $xpath) {
        $tag = $xpath->lastChild;
        if ($tag){
            $cod = $tag->getAttribute('CODIGO');
            $cont=$cod+1;
            return($cont);
        }
        else{
            return(1);
        }
    } //fim foreach
} //fim função

function listaltens($codido){
    $items = $this->xml->doc->getElementsByTagName('ENTIDADE');
    foreach ($items as $item) {
        if($item->getAttribute('ATIVO')== "S" && $codido!=$item->getAttribute('CODIGO')){
            echo "<option value=\"" . $item->getAttribute('CODIGO') . "\">" . $item->getAttribute('ID') . " - " . $item->getAttribute('CODIGO') . "</option>";
        }
    }
}

function agrupa($vetCodigo){
    $fase = new clsDados("../xml/XMLDados/" . $_SESSION['nmEntidade'] . ".xml");
    if($fase->fFaseAgrupa($vetCodigo,$this->codEnt,'Agrupamento')){
        $aux1="1";
    }
    $origem = new clsDados("../xml/XMLDados/" . $_SESSION['nmEntidade'] . ".xml");
    if($origem->fOrigemAgrupa($vetCodigo,$this->codEnt,'Agrupamento')){
        $aux2="1";
    }
    if($aux1=="1" && $aux2=="1"){return true;}
}

function desagrupa($vetCodigo){
    $fase = new clsDados("../xml/XMLDados/" . $_SESSION['nmEntidade'] . ".xml");
    if($fase->fFaseAgrupa($vetCodigo,$this->codEnt,'Desagrupamento')){
        $aux1="1";
    }
    $origem = new clsDados("../xml/XMLDados/" . $_SESSION['nmEntidade'] . ".xml");
}

```

```

        if($origem->fOrigemAgrupa($vetCodigo,$this->codEnt,'Desagrupamento')){
            $aux2="1";
        }
        if($aux1=="1" && $aux2=="1"){return true;}
    }

    function fFaseAgrupa($vetCodigo,$codEnt,$tipoAgrupamento){
        $arrayDados = explode(",",$vetCodigo);
        $cont = count($arrayDados);
        $codF = $this-
>getCodigo("/ENTIDADES/ENTIDADE[@CODIGO=".$vetCodigo[$x]."/FASES");
        $this->child->fTag("FASE","/");
        $this->child->fAtributo('CODIGO',$codF);
        $this->child->fAtributo('ATIVO','S');
        $this->child->fAtributo('tipo','Fase');
        $this->child->fAtributo('ID',$tipoAgrupamento);

        $arrayEnt = explode(",",$codEnt);
        $cont2 = count($arrayEnt);
        for ($x=0;$x<$cont2;$x++){
            $this->child->fTag("CAMPO",$arrayEnt[$x]);
            $this->child->fAtributoF('NOME',$_SESSION['nmEntidade']. " ".$arrayEnt[$x]);
        }
        for ($x=0;$x<=$cont;$x++){
            $xpath = new DOMXPath($this->xml->doc); //a variável xpath recebe um objeto
DOMXPath
            $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$arrayDados[$x]."/]";
            $path = $xpath->query($busca);
            foreach ($path as $xpath) {
                $xpath->setAttribute("ATIVO","N");
            }

            $busca = "/ENTIDADES/ENTIDADE[@CODIGO=".$arrayDados[$x]."/FASES";
            $this->tipo = "FASE";
            $this->fInclui($busca) ;
        }
        return true;
    }
}

function fOrigemAgrupa($vetCodigo,$codEnt,$tipoAgrupamento)
{
    $arrayDados = explode(",",$vetCodigo);
    $cont = count($arrayDados);
    $this->setCodEnt($codEnt);
    $this->setTipo("Origem");
    $this->setNome($tipoAgrupamento);
    $this->fOrigem("");
    for ($x=0;$x<$cont;$x++){
        $this->child->fTag("CAMPO",$arrayDados[$x]);
        $this->child->fAtributoF('NOME',$_SESSION['nmEntidade']. " ".$arrayDados[$x]);
    }

    $arrayEnt = explode(",",$codEnt);
    $cont2 = count($arrayEnt);

```

```

        for ($x=0;$x<$cont2;$x++){
            $busca = "/ENTIDADES/ENTIDADE[@CODIGO='".$arrayEnt[$x]."/ORIGEM";
            $busca1 = "/ENTIDADES/ENTIDADE[@CODIGO='".$arrayEnt[$x]."/";
            $this->fSobrescreve($busca,$busca1);
        }
        return true;
    }
}
} // fim da classe
?>

```

Classe: TransformaXML

Descrição: esta classe gerencia a transformação de arquivos XML em telas e relatórios.

```

<?
include_once("clsXML.php");

class clsConverte
{
    private $split = "<xsl:template name='split'>
        <xsl:param name='valores' />
        <xsl:variable name='valor' select='substring-before(\$valores,',';')' />
        <xsl:variable name='resto' select='substring-after(\$valores,',';')' />
        <xsl:param name='liga' />
        <xsl:variable name='valor_liga' select='substring-before(\$liga,',';')' />
        <xsl:variable name='resto_liga' select='substring-after(\$liga,',';')' />
        <xsl:if test='\$valor'>
            <xsl:if test='\$valor_liga = 1'>
                <xsl:value-of select='\$valor' />.
            </xsl:if>
        </xsl:if>
        <xsl:if test='\$resto'>
            <xsl:call-template name='split'>
                <xsl:with-param name='valores' select='\$resto' />
                <xsl:with-param name='liga' select='\$resto_liga' />
            </xsl:call-template>
        </xsl:if>
    </xsl:template>";

    private $splitEd = "<xsl:template name='split'>
        <xsl:param name='valores' />
        <xsl:variable name='valor' select='substring-before(\$valores,',';')' />
        <xsl:variable name='resto' select='substring-after(\$valores,',';')' />
        <xsl:param name='liga' />
        <xsl:variable name='valor_liga' select='substring-before(\$liga,',';')' />
        <xsl:variable name='resto_liga' select='substring-after(\$liga,',';')' />
        <xsl:if test='\$valor'>
            <xsl:value-of select='\$valor' />
            <input type='checkbox' />
            <xsl:attribute name='name' />
                <xsl:value-of select='\$valor' />
            </xsl:attribute>
            <xsl:attribute name='value' />
                <xsl:value-of select='\$valor' />
        </xsl:if>
    </xsl:template>";

```

```

        </xsl:attribute>
        <xsl:if test="\$valor_liga = 1">
            <xsl:if test="\$valor = \$valor_liga"/>
            <xsl:attribute name="\\"CHECKED\\">true</xsl:attribute >
        </xsl:if>
    </input>
</xsl:if>
<xsl:if test="\$resto">
<xsl:call-template name='split'>
    <xsl:with-param name='valores' select='\$resto'/>
    <xsl:with-param name='liga' select='\$resto_liga'/>
</xsl:call-template>
</xsl:if>
</xsl:template>
<xsl:template name='splitSelect'>
<xsl:param name='valores' />
<xsl:variable name='valor' select='substring-before(\$valores, \";\")' />
<xsl:variable name='resto' select='substring-after(\$valores, \";\")' />
<xsl:param name='liga' />
<xsl:variable name='valor_liga' select='\$liga' />
    <xsl:if test="\$valor">
        <option>
            <xsl:attribute name="\\"value\\">
                <xsl:value-of select='\$valor' />
            </xsl:attribute >
            <xsl:if test="\$valor = \$valor_liga">
                <xsl:attribute name="\\"SELECTED\\">true</xsl:attribute >
            </xsl:if>
            <xsl:value-of select='\$valor' />
        </option>
    </xsl:if>
<xsl:if test="\$resto">
    <xsl:call-template name='splitSelect'>
        <xsl:with-param name='valores' select='\$resto' />
        <xsl:with-param name='liga' select='\$valor_liga' />
    </xsl:call-template>
</xsl:if>
</xsl:template>";

```

```

private \$choose = "<xsl:choose>
    <xsl:when test="\\"@TPHTML = 'checkbox' \\">
        <xsl:call-template name=\\"split\\">
            <xsl:with-param name=\\"valores\\" select=\\"@VALORES\\" />
            <xsl:with-param name=\\"liga\\" select=\\".\\" />
        </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
        <xsl:value-of select=\\".\\" />
    </xsl:otherwise>
</xsl:choose>";

```

```

function fGeraRelatorio(\$cod1, \$cod2, \$cod3)
{
    //Mostra todos os processos de uma determina fase

```

```

if(($cod1!="all")&&($cod2!="all")&&($cod2!="origem")&&($cod3=="all")){
  $xsl= "<?xml version='1.0' encoding='ISO-8859-1'?>
  <xsl:template match='/'>
  <html>
  <head>
  <title>Relatório de Entidades</title>
  </head>
  <body>
  <br/>
  <table border='0' align='center' width='100%' cellspacing='0'>
  <xsl:for-each select='ENTIDADES/ENTIDADE'>
  <xsl:if test='@CODIGO = " . $cod1 ." '> <!--if 01-->
  <!-- faz um laço para ler cada fase-->
  <xsl:for-each select='FASES/FASE'>
  <xsl:if test='@CODIGO = " . $cod2 ." '> <!--if 02-->
  <tr bgcolor='#339933'>
  <th align='left'>
  <font color='#FFFFFF' face='Verdana, Arial, Helvetica, sans-serif'><b>
  <xsl:number value='position()' format='1.' />Fase</b> -
  <xsl:for-each select='.'>
  <xsl:value-of select='@ID' /></xsl:for-each></font></th>
  <th><font size='1' color='#FFFFFF'
  face='Verdana, Arial, Helvetica, sans-serif'>
  <b><div align='right'>
  <xsl:value-of select='../@ID' /> nº:
  <xsl:value-of select='../@CODIGO' />
  </div></b></font></th>
  </tr>
  <tr><td colspan='2' align='right'>
  <font color='#339966' size='2' face='Arial, Helvetica, sans-serif'><B>
  NOVO PROCESSO: |
  <xsl:for-each select='PROCESSOS/PROC_ADD'>
  <a><xsl:attribute
  name='href'>tela.php?e=". $_REQUEST['e']."&amp;tt=Processo&amp;cod=<xsl:value-of
  select='@CODIGO' />&amp;cod1=". $cod1."&amp;cod2=". $cod2."&amp;cod3=0</xsl:attribute><xsl:value
  of select='@NOME' /></a> |
  </xsl:for-each>
  </B></font></td></tr>
  <!-- faz um laço para ler cada processo-->
  <xsl:for-each select='PROCESSOS/PROCESSO'>
  <tr><th colspan='2' align='left'><hr size='1'
  color='green' /><font color='#339933' face='Verdana, Arial, Helvetica, sans-serif'><b>Processo
  <xsl:number value='position()' format='1' />: </b>
  <xsl:for-each select='.'>
  <xsl:value-of select='@ID' />
  </xsl:for-each></font><hr size='1'
  color='green' /></th></tr>
  <!-- lê cada campo dentro do processo-->
  <tr><td colspan='2'><xsl:for-each
  select='CAMPO'>
  <br/><font color='green'><xsl:value-of
  select='@NOME' />: </font>". $this->choose." <br/>
  </xsl:for-each><br/></td></tr>
  </xsl:for-each> <!--fim do for each processos/processo-->

```

```

        </xsl:if> <!--if 02-->
        </xsl:for-each><!--fim do for each
ENTIDADES/ENTIDADE/FASES/FASE-->
        </xsl:if> <!--if 01-->
        </xsl:for-each> <!--fim for each entidades/entidade-->
    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>";
}

elseif(($co1!="all")&&($cod2=="all")){ //Mostra o conteúdo de todas as fases

$xml= "<?xml version='1.0' encoding='ISO-8859-1'?>
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>". $this->split."
<xsl:template match='^'>
<html>
<head>
<title>Relatório de Entidades</title>
</head>
<body>
<br/>
<table border='0' align='center' width='100%' cellspacing='0'>
<xsl:for-each select='ENTIDADES/ENTIDADE'>
<xsl:if test='@CODIGO = " . $cod1 ." '> <!--if 01-->
<tr bgcolor='#339933'><th width='80%'><font color='#FFFFFF'
face='Verdana, Arial, Helvetica, sans-serif'><b><div align='left'>FASES </div></b></font></th>
<th><font size='-1' color='#FFFFFF' face='Verdana, Arial, Helvetica,
sans-serif'><b><div align='right'><xsl:value-of select='@ID' /> nº: <xsl:value-of
select='@CODIGO' /></div></b></font></th></tr>
<tr><td colspan='2' align='right'>
</td></tr>
<tr><td> </td></tr>
<!-- faz um laço para ler cada fase-->
<xsl:for-each select='FASES/FASE'>
<tr bgcolor='#339933'><th colspan='2'
align='left'><font color='#FFFFFF' face='Verdana, Arial, Helvetica, sans-serif'><b><xsl:number
value='position()' format='1.' />Fase</b> -
<xsl:for-each select='.'>
<xsl:value-of select='@ID' />
</xsl:for-each></font></th></tr>
<!-- lê cada campo dentro da fase-->
<tr><td><xsl:for-each select='CAMPO'>
<br/><font color='green'><xsl:value-of
select='@NOME' />: </font>". $this->choose." <br/>
</xsl:for-each><br/></td></tr>
</xsl:for-each><!--fim do for each
ENTIDADES/ENTIDADE/FASES/FASE-->
</xsl:if> <!--if 01-->
</xsl:for-each> <!--fim for each entidades/entidade-->
</table>
</body>

```

```

</html>
</xsl:template>
</xsl:stylesheet>";
}

elseif($cod2=="Entidade"){ //Mostra todo o conteúdo da entidade
$xsl= "<?xml version='1.0' encoding='ISO-8859-1'?">
  <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
    ".$this->split."
  <xsl:template match='^/'>
    <html>
    <head>
    <title>Relatório de Entidades</title>
    </head>
    <body>
      <br/>
      <table border='0' align='center' width='100%' cellspacing='0'>
        <xsl:for-each select='ENTIDADES/ENTIDADE'>
          <xsl:if test='@CODIGO = " . $cod1 ." '> <!--if 01-->
            <tr bgcolor='#339933'><td><font color='FFFFFF' face='Verdana,
Arial, Helvetica, sans-serif'><b><div align='center'>ENTIDADE -
          <xsl:value-of select='@ID' /> nº: <xsl:value-of
select='@CODIGO' /></div></b>
            </font></td></tr>
            <!-- faz um laço para ler cada campo da entidade-->
            <tr><td><xsl:for-each select='CAMPO'>
              <br/><font color='green'><xsl:value-of select='@NOME'
/>: </font>
              <!-- faz um laço para ler cada campo dentro da origem -->
              <xsl:choose>
                <xsl:when test='@nome = 'Agrupamento' '>
                  <br/><font color='#339966' size='2' face='Arial, Helvetica,
sans-serif'>Entidade de Origem: </font><a><xsl:attribute
name='href'>dados.php?e=" . $_REQUEST['e'] . "&amp;cod1=<xsl:value-of
select='.' />&amp;cod2=0&amp;cod3=0</xsl:attribute><font color='#000000' size='2' face='Arial,
Helvetica, sans-serif'><u><xsl:value-of select='@NOME' /></u></font></a><br/>
                </xsl:when>
                <xsl:otherwise>
                  <br/><font color='green'><xsl:value-of
select='@NOME' />: </font> ".$this->choose."<br/>
                </xsl:otherwise>
              </xsl:choose>
            </td></tr>
            <!-- faz um laço para ler cada fase-->
            <xsl:for-each select='FASES/FASE'>

```

```

<tr bgcolor="#339933"><th align="left"><font
color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><xsl:number value="position()"
format="1. \"/>Fase</b> -
<xsl:for-each select="."/ >
  <xsl:value-of select="@ID"/>
</xsl:for-each></font></th></tr>
<!-- lê cada campo dentro da fase-->
<tr><td><xsl:for-each select="CAMPO">
<xsl:choose>
  <xsl:when test="../@ID ='Agrupamento' ">
    <br/><font color="#339966" size='2' face='Arial, Helvetica,
sans-serif">Entidade de Destino: </font><a><xsl:attribute
name="href">dados.php?e=".$_REQUEST['e']."&amp;cod1=<xsl:value-of
select="."/ >&amp;cod2=0&amp;cod3=0</xsl:attribute><font color="#000000" size='2' face='Arial,
Helvetica, sans-serif"><u><xsl:value-of select="@NOME"/></u></font></a><br/>
    </xsl:when>
    <xsl:otherwise>
      <br/><font color="green"><xsl:value-of
select="@NOME" />: </font> ".$this->choose."<br/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each><br/></td></tr>
<!-- faz um laço para ler cada processo-->
<xsl:for-each select="PROCESSOS/PROCESSO">
  <tr><th align="left"><hr color="green"
size="1"/><font color="#339933" face="Verdana, Arial, Helvetica, sans-serif"><b>Processo
<xsl:number value="position()" format="1"/>: </b>
  <xsl:for-each select="."/ >
    <xsl:value-of select="@ID"/>
  </xsl:for-each></font><hr color="green"
size="1"/></th></tr>
  <!--lê cada campo dentro do processo-->
  <tr><td><xsl:for-each select="CAMPO">
    <br/><font color="green"><xsl:value-of
select="@NOME" />: </font> ".$this->choose."<br/>
    </xsl:for-each><br/></td></tr>
  </xsl:for-each> <!--fim do for each processos/processo-->
</xsl:for-each><!--fim do for each
ENTIDADES/ENTIDADE/FASES/FASE-->
  <xsl:if> <!--if 01-->
  </xsl:if>
  <xsl:for-each> <!--fim for each entidades/entidade--><tr><td align="right"
id="prt"><br/><br/><a
onClick="javascript:window.open('Relatorio.php?cod1=".$cod1."&amp;cod2=Entidade&amp;cod3=0&amp
;codimp=imp','Imprimir','width=800,height=600')"><xsl:attribute name="href">#</xsl:attribute><font
color="#339966" size='2' face='Arial, Helvetica, sans-
serif"><B><U>Imprimir<br/></U></B></font></a><br/></td></tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>;
}

```

```

elseif (($cod3!=0) && ($cod3!="all")){ //Mostra o conteúdo individual de um processo
$xml= "<?xml version='1.0' encoding='ISO-8859-1'?>
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>".$this->split."
<xsl:template match='^'>
<html>
<head>
<title>Relatório de Entidades</title>
</head>
<body>
<br></br>
<table border='0' align='center' width='100%' cellspacing='0'>
<xsl:for-each select='ENTIDADES/ENTIDADE[@CODIGO = " . $cod1
."/]FASES/FASE[@CODIGO = " . $cod2 ."]/PROCESSOS/PROCESSO[@CODIGO = " . $cod3 ."]\''>
<tr bgcolor='#339933'><th><font color='FFFFFF'
face='Verdana, Arial, Helvetica, sans-serif'><b><div align='left'>Processo: <xsl:value-of
select='@ID'></div></b></font></th>
<th><font size='-1' color='FFFFFF' face='Verdana,
Arial, Helvetica, sans-serif'><b><div align='right'><xsl:value-of select='../../../../@ID'> nº: <xsl:value-
of select='../../../../@CODIGO'></div></b></font></th></tr>
<tr><td colspan='1' align='left'><font size='-1'
color='green' face='Verdana, Arial, Helvetica, sans-serif'><B>Fase - <xsl:value-of
select='../../../../@ID'></B></font></td>
<td align='right'>
<a><xsl:attribute
name='href'>tela.php?e=".$REQUEST['e']."&cod1=".$cod1."&cod2=".$cod2."&cod3=".$
cod3."</xsl:attribute><font color='#339966' size='2' face='Arial, Helvetica, sans-
serif'><B><U>EDITAR</U></B></font></a> |
<a><xsl:attribute
name='href'>scripts/scriptExcluiDados.php?e=".$REQUEST['e']."&tp=Processo&cod1=".$co
d1."&cod2=".$cod2."&cod3=".$cod3."</xsl:attribute><font color='#339966' size='2' face='Arial,
Helvetica, sans-serif'><B><U>EXCLUIR</U></B></font></a>
</td>
</tr>
<tr><td colspan='2'><xsl:for-each select='CAMPO'>
<br/><font color='green'><xsl:value-of select='@NOME'
/> </font> ".$this->choose."<br/>
</xsl:for-each><br/></td></tr>
</xsl:for-each> <!--fim for each
entidades/entidade-->
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>";
}
elseif (($cod2!="0")&&($cod3=="0")){ //Mostra o conteúdo individual de uma fase
$xml= "<?xml version='1.0' encoding='ISO-8859-1'?>
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>".$this->split."
<xsl:template match='^'>
<html>
<head>

```

```

<title>Relatório de Entidades</title>
</head>
<body>
<br></br>
<table border="0" align="center" width="100%" cellspacing="0">
<xsl:for-each select="ENTIDADES/ENTIDADE[@CODIGO = " . $cod1
."/]/FASES/FASE[@CODIGO = " . $cod2 ."]">
<tr bgcolor="#339933"><th><font color="#FFFFFF" face="Verdana,
Arial, Helvetica, sans-serif"><b><div align="left">Fase: <xsl:value-of
select="@ID"/></div></b></font></th>
<th><font size="-1" color="#FFFFFF" face="Verdana, Arial,
Helvetica, sans-serif"><b><div align="right"><xsl:value-of select="..../@ID"/> nº: <xsl:value-of
select="..../@CODIGO"/></div></b>
</font></th></tr>
<tr><td colspan="2" align="right">
<xsl:if test="@ATIVO = 'N'">
<a><xsl:attribute
name="href">tela.php?e=" . $_REQUEST['e'] . "&cod=<xsl:value-of
select="@CODIGO_TELA"/>&tt=Fase&cod1=<xsl:value-of
select="..../@CODIGO"/>&cod2=<xsl:value-of
select="@CODIGO"/>&cod3=0</xsl:attribute><font color='#339966' size='2' face='Arial, Helvetica,
sans-serif"><b><u>CADASTRAR</u></b></font></a>
</xsl:if>
<xsl:if test="@ATIVO = 'S'">
<a><xsl:attribute
name="href">tela.php?e=" . $_REQUEST['e'] . "&cod1=<xsl:value-of
select="..../@CODIGO"/>&cod2=<xsl:value-of
select="@CODIGO"/>&cod3=0</xsl:attribute><font color='#339966' size='2' face='Arial, Helvetica,
sans-serif"><b><u>EDITAR</u></b></font></a>
</xsl:if>
</td></tr>
<tr><td colspan="2">
<xsl:if test="@ATIVO = 'N'">
<font color='#339966' size='3' face='Verdana, Arial,
Helvetica, sans-serif"><b>Não há dados cadastrados.</b></font>
</xsl:if>
<xsl:for-each select="CAMPO">
<xsl:choose>
<xsl:when test="..../@ID = 'Agrupamento' ">
<br/><font color='#339966' size='2' face='Arial, Helvetica,
sans-serif">Entidade de Destino: </font><a><xsl:attribute
name="href">dados.php?e=" . $_REQUEST['e'] . "&cod1=<xsl:value-of
select="..../@CODIGO"/>&cod2=0&cod3=0</xsl:attribute><font color='#000000' size='2' face='Arial,
Helvetica, sans-serif"><u><xsl:value-of select="@NOME"/></u></font></a><br/>
</xsl:when>
<xsl:otherwise>
<br/><font color="green"><xsl:value-of
select="@NOME" />: </font> " . $this->choose. "<br/>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each><br/></td></tr>
</xsl:for-each> <!--fim for each entidades/entidade-->
</table>

```

```

        </body>
        </html>
        </xsl:template>
        </xsl:stylesheet>;
    }
    elseif ($cod2=="origem"){ //Mostra o conteúdo de uma determinada origem
        $xsl= "<?xml version=\\"1.0\\" encoding=\\"ISO-8859-1\\"?>
        <xsl:stylesheet version=\\"1.0\\"
xmlns:xsl=\\"http://www.w3.org/1999/XSL/Transform\\">.$this->split."
        <xsl:template match=\\"^\">
        <html>
        <head>
        <title>Relatório de Entidades</title>
        </head>
        <body>
        <br></br>
            <table border=\\"0\\" align=\\"center\\" width=\\"100%\\" cellspacing=\\"0\">
                <xsl:for-each select=\\"ENTIDADES/ENTIDADE[@CODIGO =\" . $cod1 .]\">
                    <tr bgcolor=\\"#339933\"><th><font color=\\"#FFFFFF\" face=\\"Verdana,
Arial, Helvetica, sans-serif\"><b><div align=\\"left\">ORIGEM - <xsl:value-of
select=\\"ORIGEM/@nome\"/></div></b></font></th>
                    <th><font size=\\"-1\" color=\\"#FFFFFF\" face=\\"Verdana, Arial,
Helvetica, sans-serif\"><b><div align=\\"right\"><xsl:value-of select=\\"@ID\"/> nº: <xsl:value-of
select=\\"@CODIGO\"/></div></b>
                    </font></th></tr>
                    <tr><td colspan=\\"2\" align=\\"right\">
                        <xsl:if test=\\"ORIGEM/@ATIVO =N\">
                            <a><xsl:attribute
name=\\"href\">tela.php?e=\\".$_REQUEST['e'].\"&amp;cod=<xsl:value-of
select=\\"ORIGEM/@CODIGO\"/>&amp;tt=Origem&amp;cod1=<xsl:value-of
select=\\"@CODIGO\"/>&amp;cod2=0&amp;cod3=0</xsl:attribute><font color=\\"#339966\" size='2'
face='Arial, Helvetica, sans-serif'><b><u>CADASTRAR</u></b></font></a>
                            </xsl:if>
                            <xsl:if test=\\"ORIGEM/@ATIVO =S\">
                                <a><xsl:attribute
name=\\"href\">tela.php?e=\\".$_REQUEST['e'].\"&amp;cod1=<xsl:value-of
select=\\"@CODIGO\"/>&amp;cod2=origem&amp;cod3=all</xsl:attribute><font color=\\"#339966\" size='2'
face='Arial, Helvetica, sans-serif'><b><u>EDITAR</u></b></font></a>
                                </xsl:if>
                            </td></tr>
                            <xsl:if test=\\"ORIGEM/@ATIVO =N\">
                                <tr><td colspan=\\"2\">
                                    <font color=\\"#339966\" size='3' face='Verdana, Arial,
Helvetica, sans-serif'><b>Não há dados cadastrados.</b></font>
                                    </td></tr>
                                </xsl:if>
                                <tr><td colspan=\\"2\">
                                    <xsl:for-each select=\\"ORIGEM/CAMPO\">
                                        <xsl:choose>
                                            <xsl:when test=\\"../@nome ='Agrupamento' \">
                                                <br/><font color=\\"#339966\" size='2' face='Arial, Helvetica,
sans-serif'>Entiidade de Origem: </font><a><xsl:attribute
name=\\"href\">dados.php?e=\\".$_REQUEST['e'].\"&amp;cod1=<xsl:value-of

```

```

select=\".\"/>&amp;cod2=0&amp;cod3=0</xsl:attribute><font color=#000000' size='2' face='Arial,
Helvetica, sans-serif'><u><xsl:value-of select=\"@NOME\"/></u></font></a><br/>
        </xsl:when>
        <xsl:otherwise>
            <br/><font color=\"green\"><xsl:value-of
select=\"@NOME\" />: </font> ".$this->choose."<br/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each><br/></td></tr>
</xsl:for-each> <!--fim for each entidades/entidade-->
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>";
}
elseif (($cod2=="0")&&($cod3=="0")){ //Mostra os dados básicos sobre a entidade
$xmlns="<?xml version='1.0' encoding='ISO-8859-1'?">
    <xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>".$this->split."
    <xsl:template match='^'>
    <html>
    <head>
    <title>Relatório de Entidades</title>
    </head>
    <body>
    <br></br>
        <table border='0' align='center' width='100%' cellspacing='0'>
            <xsl:for-each select='ENTIDADES/ENTIDADE[@CODIGO = " . $cod1 ."]'>
                <tr bgcolor='#339933'><th><font color='#FFFFFF' face='Verdana,
Arial, Helvetica, sans-serif'><b><div align='left'>ENTIDADE -
                    <xsl:value-of select='@ID'> n°: <xsl:value-of
select='@CODIGO'></div></b>
                </font></th></tr>
                    <tr>
                        <td align='right'>
                            <a><xsl:attribute
name='href'>dados.php?e=".$_REQUEST['e']."&amp;cod1=<xsl:value-of
select='@CODIGO'>&amp;cod2=Entidade</xsl:attribute><font color='#339966' size='2' face='Arial,
Helvetica, sans-serif'><B><u>RELATÓRIO COMPLETO</u></B></font></a>
                            | <a><xsl:attribute
name='href'>tela.php?e=".$_REQUEST['e']."&amp;cod1=<xsl:value-of
select='@CODIGO'>&amp;cod2=0&amp;cod3=0</xsl:attribute><font color='#339966' size='2'
face='Arial, Helvetica, sans-serif'><u><b>EDITAR</b></u></font></a>
                            | <a><xsl:attribute
name='href'>scripts/scriptExcluiDados.php?e=".$_REQUEST['e']."&amp;tp=Entidade&amp;cod1=<xsl:v
alue-of select='@CODIGO'>&amp;cod2=0&amp;cod3=0</xsl:attribute><font color='#339966' size='2'
face='Arial, Helvetica, sans-serif'><u><b>EXCLUIR</b></u></font></a>
                        </td>
                    </tr>
                </xsl:if test='@ATIVO ='S' '>
                    <tr><td>
                        <a><xsl:attribute
name='href'>agrupar.php?e=".$_REQUEST['e']."&amp;cod1=<xsl:value-of

```

```

select=@"CODIGO"/>&amp;cod2=0&amp;cod3=0</xsl:attribute><font color='#339966' size='2'
face='Arial, Helvetica, sans-serif'><u><b>Agrupar</b></u></font></a>
| <a><xsl:attribute
name="\href">desagrupar.php?e=".$_REQUEST['e']."&amp;cod1=<xsl:value-of
select=@"CODIGO"/>&amp;cod2=0&amp;cod3=0</xsl:attribute><font color='#339966' size='2'
face='Arial, Helvetica, sans-serif'><u><b>Desagrupar</b></u></font></a>
</td>
</tr>
</xsl:if>
<xsl:if test=@"@ATIVO ='N' \">
<tr><td>
<font color='#FF0000' size='3' face='Arial, Helvetica, sans-
serif'><b>Lote finalizado.</b></font>
</td></tr>
</xsl:if>
<!-- faz um laço para ler cada campo da entidade-->
<tr><td><xsl:for-each select=@"CAMPO">
<br/><font color="green"><xsl:value-of select=@"@NOME\
/>: </font> ".$this->choose." <br/>
</xsl:for-each><br/></td></tr>
</xsl:for-each> <!--fim for each entidades/entidade-->
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>";
}
if(($cod1=="all")&&($cod2=="all")&&($cod3=="all")){ //Mostra a lista de todas as entidades
$xml= "<?xml version='1.0' encoding='ISO-8859-1'?">
<xsl:stylesheet version='1.0' xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="\^">
<html>
<head>
<title>Relatório de Entidades</title>
</head>
<body>
<br></br>
<table border="0" align="center" width="100%" cellspacing="0">
<tr bgcolor="\#339933"><th><font color="\#FFFFFF" face="\Verdana,
Arial, Helvetica, sans-serif"><b><div align="left">LISTA DE ENTIDADES</div></b></font></th></tr>
<tr><td align="right"><a
href="tela.php?e=".$_REQUEST['e']."&amp;tt=Entidade\ "><font color='#339966' size='2' face='Arial,
Helvetica, sans-serif'><u><b>NOVO REGISTRO</b></u></font></a></td></tr>
<xsl:for-each select="ENTIDADES/ENTIDADE\ ">
<tr><td><br/><b><a><xsl:attribute
name="\href">dados.php?e=".$_REQUEST['e']."&amp;cod1=<xsl:value-of
select=@"CODIGO"/>&amp;cod2=Entidade</xsl:attribute><font color='#339966' size='3' face='Arial,
Helvetica, sans-serif'><xsl:value-of select=@"@ID"/> nº: <xsl:value-of
select=@"CODIGO"/></font></a></b></td></tr>
</xsl:for-each> <!--fim for each entidades/entidade-->
</table>
</body>
</html>

```

```

        </xsl:template>
        </xsl:stylesheet>";
    }

    $this->fTransfStringXML($xsl,"../xml/" .$_SESSION['nmEntidade'].".xml");
} // fim funcao fGeralatorio

//Função q converte O XML DE TELA EM UMA TELA PARA INCLUIR DADOS
function fGeraTela($codTela,$xml,$tipo,$e,$codEnt,$codFase)
{
    $ident="";
    IF ($tipo!="Entidade"){ $ident=$_SESSION["nmEntidade"]." n°: ".$codEnt;}
    $xsl = "<?xml version='1.0' encoding='ISO-8859-1'?>
        <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
            <xsl:template match='^'>
                <html>
                <head>
                <title>Cadastro de Dados</title>
                </head>
                <body>
                <br></br>
                <form name='grava' method='post'
action='scripts/scriptGavaDados.php?Grava=S&amp;e=".$e."&amp;codTela=".$codTela."&amp;codEnt="
.$codEnt. "&amp;codFase=".$codFase."'>
                <input type='hidden' name='".$tipo."'>
                    <xsl:attribute name='value'>
                        <xsl:value-of
select='banco_telas/'.strtoupper($tipo)."[@CODIGO =".$codTela."]/@nome \"/>
                    </xsl:attribute>
                </input>
                <table border='0' align='center' width='100%' cellspacing='0'>
                <tr bgcolor='#339933'><th colspan='10'><font color='#FFFFFF"
face='Verdana, Arial, Helvetica, sans-serif'><b><div align='left'>CADASTRO DE ".strtoupper($tipo)." -
<xsl:value-of select='banco_telas/'.strtoupper($tipo)."[@CODIGO =".$codTela."]/@nome"
/></div></b></font></th><th><font size='-1' color='#FFFFFF" face='Verdana, Arial, Helvetica, sans-
serif'><b><div align='right'>".$ident." </div></b></font></th></tr>
                <xsl:for-each select='banco_telas/'.strtoupper($tipo)."/campo">
                <xsl:if test='../@CODIGO =".$codTela."'>
                <tr valign='top'>
                <xsl:if test=' thtml = 'text' '>
                <td align='left' width='20%'><xsl:value-of select='nome' /></td>
                <td align='left' colspan='4'>
                <input type='text'>
                    <xsl:attribute name='name'>
                        <xsl:value-of select='@CODIGO' />
                    </xsl:attribute>
                    <xsl:attribute name='value'>
                        <xsl:value-of select='valores' />
                    </xsl:attribute>
                    <xsl:attribute name='size'>
                        <xsl:value-of select='30' />
                    </xsl:attribute>
                </input></td>

```

```

</xsl:if>
<xsl:if test=" thtml = 'textarea' \">
<td align="left" width="20%"><xsl:value-of select="nome" /></td>
<td colspan="4">
  <textarea>
    <xsl:attribute name="name">
      <xsl:value-of select="@CODIGO" />
    </xsl:attribute>
    <xsl:value-of select="valores" />
  </textarea>
</td>
</xsl:if>
<xsl:if test=" thtml = 'checkbox' \">
<td align="left" width="20%">
<input type="hidden">
  <xsl:attribute name="name">HDN0<xsl:number value="position()"
format="1" /></xsl:attribute>
  <xsl:attribute name="value"><xsl:value-of select="@CODIGO"
/></xsl:attribute>
</input>
<xsl:value-of select="nome" /></td>
<xsl:for-each select="valores/opcao">
<td width=""><xsl:value-of select="." />
<input type="checkbox">
  <xsl:attribute name="name">
    <xsl:value-of select="." />
  </xsl:attribute>
  <xsl:attribute name="value">
    <xsl:value-of select="." />
  </xsl:attribute>
</input>
</td>
</xsl:for-each>
<input type="hidden">
  <xsl:attribute name="name">HDN1<xsl:number value="position()"
format="1" /></xsl:attribute>
  <xsl:attribute name="value"><xsl:value-of select="@CODIGO"
/></xsl:attribute>
</input>
</xsl:if>
<xsl:if test=" thtml = 'select' \">
<td align="left" width="20%">
<xsl:value-of select="nome" /></td>
<td colspan="4">
  <select>
    <xsl:attribute name="name">
      <xsl:value-of select="@CODIGO" />
    </xsl:attribute>
    <xsl:for-each select="valores/opcao">
      <option>
        <xsl:attribute name="value">
          <xsl:value-of select="." />
        </xsl:attribute >
        <xsl:value-of select="." />

```

```

                </option>
            </xsl:for-each>
        </select>
    </td>
</xsl:if>
</tr>
</xsl:if>
</xsl:for-each>
</table>
<p>
    <input name="btnSalvar" type="submit" value="Salvar"></input>
</p>
</form>
</body>
</html>
</xsl:template>
</xsl:stylesheet>";
$this->fTransfStringXML($xsl,$xml);
}

function fGeraTelaEd($xml,$cod1,$cod2,$cod3,$e)
{
    if (($cod2=="0")&&($cod3=="0")){ //Monta a tela de edição de entidade
        $xsl = "<?xml version='1.0' encoding='ISO-8859-1'?>
            <xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>". $this-> splitEd."
            <xsl:template match='^/'>
                <html>
                    <head>
                        <title>Cadastro de Dados</title>
                    </head>
                    <body>
                        <br></br>
                        <form name="grava" method="post"
action="scripts/scriptAlteraDados.php?Grava=S&amp;e=".$e."&amp;cod1=".$cod1."&amp;cod2=".$cod2.
"&amp;cod3=".$cod3.">
                            <input type="hidden" name="Entidade">
<xsl:attribute name="value">
<xsl:value-of select="ENTIDADES/ENTIDADE/@ID"/>
</xsl:attribute>
                            </input>
                            <table border="0" align="center" width="100%">
                                <xsl:for-each
select="ENTIDADES/ENTIDADE[@CODIGO=".$cod1.]">
                                    <tr bgcolor="#339933">
                                        <th colspan="10">
                                            <font color="#FFFFFF"
face="Verdana, Arial, Helvetica, sans-serif"><b>
                                                <div align="left">EDITAR -
<xsl:value-of select="@ID" /> nº <xsl:value-of select="@CODIGO" /></div>
                                            </b></font>
                                        </th>
                                    </tr>
                                </xsl:for-each>
                            </table>
                        </body>
                    </html>
            </xsl:template>
        </xsl:stylesheet>";
    }
}

```

```

width="20%"><xsl:value-of select="@NOME" /></td>
name="name">
select="@CODIGOCAMPO" />
name="value">
select="." />
name="size">
select='30' />
width="20%"><xsl:value-of select="@NOME" /></td>
name="name">
select="@CODIGOCAMPO" />
/>
">
<xsl:attribute name="name">HDN0<xsl:number value="position()" format="1" /></xsl:attribute>
<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>
</input>
/></td>
<td align="left">
<xsl:call-template name="split">
<xsl:for-each select="CAMPO">
  <tr valign="top">
    <xsl:if test=" @TPHTML = 'text' ">
      <td align="left"
        <td align="left" colspan="4">
          <input type='text'
            <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
                <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
                <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
            </input>
          </td>
        </xsl:if>
        <xsl:if test=" @TPHTML = 'textarea' ">
          <td align="left"
            <td colspan="4">
              <textarea>
                <xsl:attribute
                  <xsl:value-of
                    </xsl:attribute>
                    <xsl:value-of select="." />
              </textarea>
            </td>
          </xsl:if>
          <xsl:if test=" @TPHTML = 'checkbox'
            <td align="left" width="20%">
              <input type="hidden">
<xsl:attribute name="name">HDN0<xsl:number value="position()" format="1" /></xsl:attribute>
<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>
</input>
/></td>
<td align="left">
<xsl:call-template name="split">
<xsl:for-each select="CAMPO">
  <tr valign="top">
    <xsl:if test=" @TPHTML = 'text' ">
      <td align="left"
        <td align="left" colspan="4">
          <input type='text'
            <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
                <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
                <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
            </input>
          </td>
        </xsl:if>
        <xsl:if test=" @TPHTML = 'textarea' ">
          <td colspan="4">
              <textarea>
                <xsl:attribute
                  <xsl:value-of
                    </xsl:attribute>
                    <xsl:value-of select="." />
              </textarea>
            </td>
          </xsl:if>
          <xsl:if test=" @TPHTML = 'checkbox'
            <td align="left" width="20%">
              <input type="hidden">

```

```

<xsl:with-param name="valores" select="@VALORES"/>
<xsl:with-param name="liga" select="."/ />
</xsl:call-template>

<input type="hidden">

<xsl:attribute name="name">HDN1<xsl:number value="position()" format="1" /></xsl:attribute>

<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>

</input>

</td>
</xsl:if>
<xsl:if test=" @TPHTML = 'select' ">
  <td align="left" width="20%">
    <xsl:value-of select="@NOME"

select="@CODIGOCAMPO" />

    <td colspan="4">
      <select>
        <xsl:attribute name="name">
          <xsl:value-of

        </xsl:attribute>

      </select>
    </td>
  </xsl:if>
</tr>
</xsl:for-each>
</xsl:for-each>
</table>
<p>
  <input name="btnAlterar" type="submit"

value="Alterar"></input>
</p>
</form>
</body>
</html>
</xsl:template>
</xsl:stylesheet>;
}

if (($cod2!="0")&&($cod3=="0")){ //Monta tela de edição de Fases
$xml = "<?xml version="1.0" encoding="ISO-8859-1"?>
  <xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">.$this-> splitEd."
  <xsl:template match="^">
    <html>

```

```

<head>
  <title>Cadastro de Dados</title>
</head>
<body>
  <br></br>
  <form name="grava" method="post"
action="\scripts/scriptAlteraDados.php?Grava=S&amp;e=". $e."&amp;cod1=". $cod1."&amp;cod2=". $cod2.
"&amp;cod3=". $cod3."\">
    <input type="hidden" name="Fase\">

<xsl:attribute name="value\">

<xsl:value-of select="\ENTIDADES/ENTIDADE[@CODIGO=". $cod1."]/FASES/FASE/@ID\"/>

</xsl:attribute>

                                                                </input>
    <table border="0" align="center" width="100%">
      <xsl:for-each
select="\ENTIDADES/ENTIDADE[@CODIGO=". $cod1."]/FASES/FASE[@CODIGO=". $cod2."]\>
        <tr bgcolor="#339933">
          <th colspan="10">
            <font color="FFFFFF"
face="Verdana, Arial, Helvetica, sans-serif"><b>
                                                                <div align="center">EDITAR
- <xsl:value-of select="@ID" /></div>
                                                                </b></font>
          </th>
        </tr>
        <xsl:for-each select="CAMPO">
          <tr valign="top">
            <xsl:if test="@TPHTML = 'text' \">
              <td align="left"
width="20%"><xsl:value-of select="@NOME" /></td>
              <td align="left" colspan="4">
                <input type="text">
                  <xsl:attribute
                                                                <xsl:value-of
select="@CODIGOCAMPO" />
                  </xsl:attribute>
                  <xsl:attribute
                                                                <xsl:value-of
name="value\">
select=".\" />
                  </xsl:attribute>
                  <xsl:attribute
                                                                <xsl:value-of
name="size\">
select='30' />
                  </xsl:attribute>
                </td>
              </xsl:if>

```

```

width="20%"><xsl:value-of select="@NOME" />:</td>
name="name">
select="@CODIGOCAMPO" />
/>
">
<xsl:attribute name="name">HDN0<xsl:number value="position()" format="1" /></xsl:attribute>
<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>
</input>
/>:</td>
<xsl:call-template name="split">
<xsl:with-param name="valores" select="@VALORES"/>
<xsl:with-param name="liga" select="."/>
</xsl:call-template>
<xsl:attribute name="name">HDN1<xsl:number value="position()" format="1" /></xsl:attribute>
<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>
</input>
/>:</td>
select="@CODIGOCAMPO" />

```

```

<xsl:if test=" @TPHTML = 'textarea' ">
<td align="left"
<td colspan="4">
<textarea>
<xsl:attribute
<xsl:value-of
</xsl:attribute>
<xsl:value-of select=".">
</textarea>
</td>
</xsl:if>
<xsl:if test=" @TPHTML = 'checkbox'
<td align="left" width="20%">
<input type="hidden">
<xsl:value-of select="@NOME"
<td align="left">
<input type="hidden">
</td>
</xsl:if>
<xsl:if test=" @TPHTML = 'select' ">
<td align="left" width="20%">
<xsl:value-of select="@NOME"
<td colspan="4">
<select>
<xsl:attribute name="name">
<xsl:value-of

```

```

</xsl:attribute>

<xsl:call-template name="splitSelect">
  <xsl:with-param name="valores" select="@VALORES"/>
  <xsl:with-param name="liga" select="."/>
</xsl:call-template>
</select>
</td>
</xsl:if>
</tr>
</xsl:for-each>
</xsl:for-each>
</table>
<p>
  <input name="btnAlterar" type="submit"
value="Alterar"></input>
</p>
</form>
</body>
</html>
</xsl:template>
</xsl:stylesheet>;
}

if($cod2=="origem"){ // Monta a tela de origem
  $xsl = "<?xml version='1.0' encoding='ISO-8859-1'?>
  <xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>". $this-> splitEd."
  <xsl:template match='/'>
  <html>
  <head>
  <title>Cadastro de Dados</title>
  </head>
  <body>
  <br></br>
  <form name="grava" method="post"
action="scripts/scriptAlteraDados.php?Grava=S&amp;e=".$e."&amp;cod1=".$cod1."&amp;cod2=".$cod2.
"&amp;cod3=".$cod3.">
  <input type="hidden" name="Origem">

<xsl:attribute name="value">
<xsl:value-of select="ENTIDADES/ENTIDADE[@CODIGO=".$cod1."]/ORIGEM/@nome"/>
</xsl:attribute>
</input>
<table border="0" align="center" width="100%"
cellspacing="0">
  <xsl:for-each
select="ENTIDADES/ENTIDADE[@CODIGO=".$cod1."]/ORIGEM">
  <tr bgcolor="#339933">

```

```

face="Verdana, Arial, Helvetica, sans-serif"><b>
ORIGEM - <xsl:value-of select="@nome" /></div>

<th colspan="10">
  <font color="#FFFFFF"
    <div align="left">EDITAR
  </b></font>
</th>

<th><font size="-1" color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b><div
align="right">".$_SESSION["nmEntidade"]." nº: ".$cod1."</div></b></font></th>
</tr>
<xsl:for-each select="CAMPO">
  <tr valign="top">
    <xsl:if test="@TPHTML = 'text' ">
      <td align="left"
        width="20%"><xsl:value-of select="@NOME" /></td>
        <td align="left" colspan="4">
          <input type="text">
            <xsl:attribute
              <xsl:value-of
                </xsl:attribute>
              <xsl:attribute
                <xsl:value-of
                  </xsl:attribute>
                <xsl:attribute
                  <xsl:value-of
                    </xsl:attribute>
                  </input>
                </td>
              </xsl:if>
            <xsl:if test="@TPHTML = 'textarea' ">
              <td align="left"
                width="20%"><xsl:value-of select="@NOME" /></td>
                <td colspan="4">
                  <textarea>
                    <xsl:attribute
                      <xsl:value-of
                        </xsl:attribute>
                      <xsl:value-of select=".">
                    </td>
                  </xsl:if>
                <xsl:if test="@TPHTML = 'checkbox'
                  <td align="left" width="20%">

```

```

<input type="hidden">
<xsl:attribute name="name">HDN0<xsl:number value="position()" format="1" /></xsl:attribute>
<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>
</input>
/>:</td>

```

```

<xsl:value-of select="@NOME"
<td align="left">

```

```

<xsl:call-template name="split">
<xsl:with-param name="valores" select="@VALORES"/>
<xsl:with-param name="liga" select="."/>
</xsl:call-template>

```

```

<input type="hidden">

```

```

<xsl:attribute name="name">HDN1<xsl:number value="position()" format="1" /></xsl:attribute>
<xsl:attribute name="value"><xsl:value-of select="@CODIGOCAMPO" /></xsl:attribute>
</input>

```

```

/>:</td>

```

```

</td>
</xsl:if>
<xsl:if test="@TPHTML = 'select' ">
  <td align="left" width="20%">
    <xsl:value-of select="@NOME"

  <td colspan="4">
    <select>
      <xsl:attribute name="name">
        <xsl:value-of

    </xsl:attribute>

```

```

select="@CODIGOCAMPO" />

```

```

<xsl:call-template name="splitSelect">
<xsl:with-param name="valores" select="@VALORES"/>
<xsl:with-param name="liga" select="."/>
</xsl:call-template>

```

```

</select>
</td>
</xsl:if>
</tr>
</xsl:for-each>
</xsl:for-each>
</table>
<p>

```

```

value=\Alterar\"></input>
</p>
</form>
</body>
</html>
</xsl:template>
</xsl:stylesheet>;
}

if (($cod3!=0) && ($cod3!="all")){ // Monta a tela de edição de processos
$xml = "<?xml version=\1.0\ encoding=\ISO-8859-1\?>
<xsl:stylesheet version=\1.0\
xmlns:xsl=\http://www.w3.org/1999/XSL/Transform\>". $this-> splitEd."
<xsl:template match=\^/\>
<html>
<head>
<title>Cadastro de Dados</title>
</head>
<body>
<br></br>
<form name=\grava\ method=\post\
action=\scripts/scriptAlteraDados.php?Grava=S&amp;e=.\$.&amp;cod1=.\$cod1.&amp;cod2=.\$cod2.
&amp;cod3=.\$cod3.\>
<input type=\hidden\ name=\Processo\>

<xsl:attribute name=\value\>

<xsl:value-of
select=\ENTIDADES/ENTIDADE[@CODIGO=.\$cod1.\]/FASES/FASE[@CODIGO=.\$cod2.\]/PROCES
SOS/PROCESSO/@ID\/>

</xsl:attribute>

</input>
<table border=\0\ align=\center\ width=\100%\>
<xsl:for-each
select=\ENTIDADES/ENTIDADE[@CODIGO=.\$cod1.\]/FASES/FASE[@CODIGO=.\$cod2.\]/PROCES
SOS/PROCESSO[@CODIGO=.\$cod3.\]\>
<tr bgcolor=\#339933\>
<th colspan=\10\>
<font color=\#FFFFFF\
face=\Verdana, Arial, Helvetica, sans-serif\><b>
<div align=\center\>Editar
Processo: <xsl:value-of select=\@ID\ /> nº <xsl:value-of select=\@CODIGO\ /></div>
</b></font>
</th>
</tr>
<xsl:for-each select=\CAMPO\>
<tr valign=\top\>
<xsl:if test=\ @TPHTML = 'text' \>
<td align=\left\
width=\20%\><xsl:value-of select=\@NOME\ /></td>

<td align=\left\ colspan=\4\>
<input type='text'>

```

```

name="\name\ ">
select="\@CODIGOCAMPO" />

name="\value\ ">
select=".\ " />

name="\size\ ">
select='30' />

width="20\ "><xsl:value-of select="\@NOME\ " /></td>

name="\name\ ">
select="\@CODIGOCAMPO" />

/>

\ ">

<xsl:attribute name="\name\ ">HDN0<xsl:number value="\position()" format="1\ " /></xsl:attribute>
  <xsl:attribute name="\value\ "><xsl:value-of select="\@CODIGOCAMPO" /></xsl:attribute>
  </input>
  <xsl:value-of select="\@NOME\ " /></td>
  <td align="left">
    <xsl:call-template name="split">
      <xsl:with-param name="valores\ " select="\@VALORES\ "/>
      <xsl:with-param name="liga\ " select=".\ "/>
    </xsl:call-template>
    <input type="hidden\ ">
<xsl:attribute name="\name\ ">HDN1<xsl:number value="\position()" format="1\ " /></xsl:attribute>
<xsl:attribute name="\value\ "><xsl:value-of select="\@CODIGOCAMPO\ " /></xsl:attribute>
  </input>
  </td>
</xsl:if>
<xsl:if test="\ @TPHTML = 'select' \ ">
  <td align="left" width="20\ ">

```

```

        <xsl:value-of select="\@NOME\" />.</td>
        <td colspan="4">
        <select>
            <xsl:attribute name="name">
        <xsl:value-of select="\@CODIGOCAMPO\" />
            </xsl:attribute>
            <xsl:call-template name="splitSelect">
        <xsl:with-param name="valores\" select="\@VALORES\"/>
            <xsl:with-param name="liga\" select="\"./>
            </xsl:call-template>
        </select>
        </td>
        </xsl:if>
        </tr>
    </xsl:for-each>
</table>
<p>
    <input name="btnAlterar\" type="submit"
value="Alterar"></input>
</p>
</form>
</body>
</html>
</xsl:template>
</xsl:stylesheet>";
    }
    $this->fTransfStringXML($xsl,$xml);
}

/*****Funções responsáveis pela transformação de um arquivo XML através de um XSL*****/
function fTransfStringXML($xsl,$xml)
{
    $xslt = new xsltProcessor;
    $xslt->importStyleSheet(DomDocument::loadXML($xsl));
    print $xslt->transformToXML(DomDocument::load($xml));
}

function fTransformaXML($xsl,$xml)
{
    $xslt = new xsltProcessor();
    $xslt->importStyleSheet(DomDocument::load($xsl));
    print $xslt->transformToXML(DomDocument::load($xml));
}
} //fim classe
?>

```