

UNIVERSIDADE FEDERAL DO PARANÁ

DANIELLE FORBECI SUZUKI

MAYKO BORRET OLIVEIRA

GERENCIADOR DE FORMULÁRIOS ELETRÔNICOS

CURITIBA

2010

DANIELLE FORBECI SUZUKI

MAYKO BORRET OLIVEIRA

## GERENCIADOR DE FORMULÁRIOS ELETRÔNICOS

Relatório apresentado como requisito parcial para conclusão do Curso Superior de Tecnologia em Sistemas de Informação, do Setor de Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná.

Orientador: Professor Dieval Guizelini.

Co-orientadora: Professora Jeroniza Marchaukoski.

CURITIBA

2010

Termo de Aprovação

DANIELLE FORBECI SUZUKI

MAYKO BORRET OLIVEIRA

GERENCIADOR DE FORMULÁRIOS ELETRÔNICOS

Trabalho aprovado como requisito para obtenção do grau de Tecnólogo em Sistemas de Informação, no curso de Graduação em Tecnologia em Sistemas de Informação, da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador: \_\_\_\_\_

Prof. Msc. Dieval Guizelini

Co-orientadora: \_\_\_\_\_

Prof<sup>a</sup>. Dra. Jeroniza Marchaukoski

\_\_\_\_\_  
Prof. Dr. Roberto Tadeu Raittz

## AGRADECIMENTOS

À Deus, por ter iluminado nossos caminhos durante essa longa jornada. Às nossas famílias, pelo apoio e compreensão nas horas difíceis. A todos os professores do curso de Tecnologia em Sistemas de Informação, pelos ensinamentos e experiências passados. Gostaríamos também de deixar registrado nesse trabalho, nosso profundo agradecimento ao nosso orientador, MSc. Dieval Guizelini, o qual forneceu todo subsídio e incentivo necessário para que concluíssemos este trabalho. Aos colegas do curso pelo companheirismo, integração e colaboração. Em particular à colega que desistiu de nossa equipe, pois nos deu a oportunidade de provar mais uma vez que somos capazes. Aos mestres pelo trabalho de pesquisa e desenvolvimento, tornando acessível e ampliando o conhecimento e a tecnologia, sem estes seria muito difícil a realização deste projeto.

## RESUMO

O presente trabalho descreve a modelagem e a implementação de um Sistema web para gerenciamento de formulários dinâmicos desenvolvido pelos alunos do Curso de Tecnologia em Sistemas de Informação da UFPR, com a finalidade principal de auxiliar o setor do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da UFPR, a gerir de uma melhor forma possível todos os processos pertinentes a cadastros e criação de novos formulários no processo acadêmico. Proporcionando uma maior organização, armazenamento, segurança e disponibilidade dos dados pertinentes a cada formulário. O Sistema foi desenvolvido utilizando a linguagem de programação PHP e base de dados MYSQL. Um dos aspectos mais importantes do software que não podemos deixar de destacar está diretamente ligado a flexibilidade das ferramentas utilizadas na construção, podendo ser customizado de acordo com os interesses do usuário e da organização.

Palavras-chave: Aplicação Web, PHP, MYSQL e Formulários dinâmicos.

## ABSTRACT

This paper describes the modeling and implementation of a web system for managing dynamic forms developed by the students of Technology in Information Systems, UFPR, with the primary purpose of assisting the industry in the Course Technology Analysis and Development Systems UFPR, managing a best possible way all the relevant records and processes creating new forms in the academic process. Providing a better organization, storage, security and availability of data pertinent to each form. O system was developed using the PHP programming language and MySQL database. One of the most important aspects of software that can not fail to mention is directly linked to the flexibility of the tools used in construction and can be customized to specific user interests and the organization.

Keywords: Web Application, PHP, MYSQL and dynamic forms.

## SUMÁRIO

<b>LISTA DE SIGLAS.....</b>	<b>11</b>
<b>1. INTRODUÇÃO .....</b>	<b>13</b>
1.1. INTRODUÇÃO .....	13
1.2. OBJETIVOS DO PROJETO .....	14
1.2.1. OBJETIVO GERAL .....	15
1.2.2. OBJETIVOS ESPECÍFICOS.....	15
1.2.3. JUSTIFICATIVA.....	15
<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>16</b>
2.1. REVISÃO BIBLIOGRÁFICA .....	16
2.1.1. LINGUAGEM E AMBIENTE DE DESENVOLVIMENTO .....	17
2.1.2. ARQUITETURA DE SISTEMAS WEB .....	18
2.1.3. FORMULÁRIOS ESTÁTICOS E DINÂMICOS .....	20
2.2. FUNCIONALIDADES E VANTAGENS .....	21
2.3. FUNCIONALIDADES GERAIS.....	22
2.3.1. ADMINISTRADOR DO SISTEMA.....	22
2.3.2. USUÁRIO DO SISTEMA.....	23
<b>3. METODOLOGIA.....</b>	<b>23</b>
3.1. METODOLOGIA DO TRABALHO .....	23
3.1.1. ORIENTAÇÃO A OBJETO.....	23
3.1.2. UML – INNIFIED MODELING LANGUAGE .....	24
3.1.3. CARACTERÍSTICAS DA UML.....	25
3.1.4. APLICAÇÕES DA UML.....	27
3.1.4.1. FASE DO DESENVOLVIMENTO DE UM SISTEMA EM UML.....	28
3.1.4.2. ANÁLISE DE REQUISITOS .....	28
3.1.4.3. ANÁLISE .....	29
3.1.4.4. PROJETO .....	29
3.1.4.5. IMPLEMENTAÇÃO (PROGRAMAÇÃO) .....	30
3.1.4.6. TESTE.....	31
3.1.4.7. DIAGRAMA DE CASOS USO .....	31
3.1.4.8. DIAGRAMA DE CLASSES.....	33
3.1.5. ARQUITETURA EM CAMADAS .....	37
3.1.6. DESIGN PATTERNS .....	39

3.1.7.	MVC - MODEL VIEW CONTROLLER.....	40
3.1.8.	APLICAÇÃO DA METODOLOGIA “METADADOS” .....	40
3.2.	TECNOLOGIA E SOFTWARE .....	41
3.2.1.	ESPECIFICAÇÃO E MODELAGEM UML DO SISTEMA.....	41
3.2.1.1.	JUDE .....	41
3.2.2.	LINGUAGEM E AMBIENTE DE DESENVOLVIMENTO DO SISTEMA .....	42
3.2.2.1.	PHP .....	42
3.2.2.2.	NETBEANS .....	42
3.2.2.3.	MYSQL.....	42
3.2.3.	CAMADA DE APRESENTAÇÃO DO SISTEMA .....	42
3.2.3.1.	HTML.....	42
3.2.3.2.	CSS.....	43
3.2.3.3.	JQUERY.....	43
3.2.3.4.	JAVASCRIPT .....	43
3.2.3.5.	ASYNCHRONOUS JAVASCRIPT AND XML – AJAX.....	44
3.2.4.	CAMADA DE CONTROLE DO SISTEMA.....	45
3.2.5.	CAMADA DE MODELO E DE PERSISTÊNCIA DO SISTEMA.....	45
3.2.6.	REESCRITA DE URL .....	45
3.2.7.	FERRAMENTAS DE INFRAESTRUTURA DO SISTEMA .....	46
3.2.7.1.	APACHE.....	46
3.2.7.2.	SUBVERSION .....	46
3.2.7.3.	UNFUDDLE .....	46
3.3.	DESENVOLVIMENTO DO PROJETO .....	46
3.3.1.	ESTUDO DE CASO – GERENCIADOR DE FORMULÁRIOS ELETRÔNICOS.....	48
<b>4.</b>	<b>APRESENTAÇÃO DO SOFTWARE .....</b>	<b>48</b>
4.1.	DIAGRAMA DE CASO DE USO.....	49
4.2.	ESPECIFICAÇÃO DOS CASOS DE USO .....	49
4.2.1.	CASO DE USO: MANTER FORMULÁRIO .....	49
4.2.1.1.	TELAS .....	54
4.2.1.2.	DIAGRAMA DE SEQUENCIA: INCLUIR FORMULÁRIO .....	59
4.2.1.3.	DIAGRAMA DE SEQUENCIA: ALTERAR FORMULÁRIO .....	59
4.2.2.	CASO DE USO: MANTER CAMPO .....	59



4.2.2.1.	TELAS .....	62
4.2.2.2.	DIAGRAMA DE SEQUENCIA: INCLUIR CAMPO .....	64
4.2.2.3.	DIAGRAMA DE SEQUENCIA : ALTERAR CAMPO .....	64
4.2.2.4.	DIAGRAMA DE SEQUENCIA: EXCLUIR CAMPO .....	64
4.2.3.	CASO DE USO: GERAR RELATÓRIO .....	64
4.2.3.1.	DIAGRAMA DE SEQUENCIA: GERAR RELATÓRIO .....	66
4.2.4.	CASO DE USO: MANTER TIPO DE CAMPO .....	66
4.2.4.1.	TELAS .....	69
4.2.4.2.	DIAGRAMA DE SEQUENCIA: INCLUIR TIPO DE CAMPO .....	71
4.2.4.3.	DIAGRAMA DE SEQUENCIA: ALTERAR TIPO DE CAMPO .....	71
4.2.4.4.	DIAGRAMA DE SEQUENCIA: EXCLUIR TIPO DE CAMPO .....	71
4.2.5.	CASO DE USO: MANTER TIPO DE DADO .....	71
4.2.5.1.	TELAS .....	75
4.2.5.2.	DIAGRAMA DE SEQUENCIA: INCLUIR TIPO DE DADO .....	77
4.2.5.3.	DIAGRAMA DE SEQUENCIA: ALTERAR TIPO DE DADO .....	77
4.2.5.4.	DIAGRAMA DE SEQUENCIA: EXCLUIR TIPO DE DADO .....	77
4.2.6.	CASO DE USO: PREENCHER FORMULÁRIO .....	77
4.2.6.1.	TELAS .....	79
4.2.6.2.	DIAGRAMA DE SEQUENCIA: PREENCHER FORMULÁRIO .....	81
4.2.7.	CASO DE USO: EFETUAR LOGIN .....	81
4.2.7.1.	TELAS .....	83
4.2.7.2.	DIAGRAMA DE SEQUENCIA: LOGIN .....	84
4.2.8.	CASO DE USO: VISUALIZAR QUESTIONÁRIOS .....	84
4.2.8.1.	TELAS .....	86
4.2.8.2.	DIAGRAMA DE SEQUENCIA: LOGIN .....	88
4.3.	DIAGRAMA DE CLASSES .....	88
4.4.	BANCO DE DADOS .....	88
4.4.1.	DIAGRAMA DE ENTIDADE-RELACIONAMENTO .....	88
4.4.2.	DICIONÁRIO DE DADOS .....	88
4.5.	DEPENDÊNCIAS DO SISTEMA .....	88
<b>5.</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>88</b>
<b>5.1.</b>	<b>IMPLEMENTAÇÕES FUTURAS .....</b>	<b>89</b>
	<b>REFERÊNCIAS.....</b>	<b>91</b>

<b>BIBLIOGRAFIA .....</b>	<b>93</b>
<b>APÊNDICES .....</b>	<b>94</b>
APÊNDICE A – DIAGRAMA DE CASO DE USO .....	94
APÊNDICE B – DIAGRAMA DE SEQUÊNCIA: INCLUIR FORMULÁRIO .....	95
APÊNDICE C – DIAGRAMA DE SEQUÊNCIA: ALTERAR FORMULÁRIO .....	96
APÊNDICE D – DIAGRAMA DE SEQUÊNCIA: INCLUIR CAMPO .....	97
APÊNDICE E – DIAGRAMA DE SEQUÊNCIA: ALTERAR CAMPO .....	98
APÊNDICE F – DIAGRAMA DE SEQUÊNCIA: EXCLUIR CAMPO .....	99
APÊNDICE G – DIAGRAMA DE SEQUÊNCIA: GERAR RELATÓRIO .....	100
APÊNDICE H – DIAGRAMA DE SEQUÊNCIA: INCLUIR TIPO DE CAMPO .....	101
APÊNDICE I – DIAGRAMA DE SEQUÊNCIA: ALTERAR TIPO DE CAMPO .....	102
APÊNDICE J – DIAGRAMA DE SEQUÊNCIA: EXCLUIR TIPO DE CAMPO .....	103
APÊNDICE K – DIAGRAMA DE SEQUÊNCIA: INCLUIR TIPO DE DADO .....	104
APÊNDICE L – DIAGRAMA DE SEQUÊNCIA: ALTERAR TIPO DE DADO .....	105
APÊNDICE M – DIAGRAMA DE SEQUÊNCIA: EXCLUIR TIPO DE DADO .....	106
APÊNDICE N – DIAGRAMA DE SEQUÊNCIA: PREENCHER FORMULÁRIO .....	107
APÊNDICE O – DIAGRAMA DE SEQUÊNCIA: LOGIN .....	108
APÊNDICE P – DIAGRAMA DE CLASSES .....	109
APÊNDICE Q – DIAGRAMA ENTIDADE-RELACIONAMENTO .....	110
APÊNDICE R – DICIONÁRIO DE DADOS .....	111

## LISTA DE SIGLAS

API	–	Application Programming Interface
CSS	–	Cascading Style Sheet
HTML	–	HyperText Markup Language
JUDE	–	Java and UML Developer Environment
MVC	–	Model-View-Controller
SGBD	–	Sistema Gerenciador de Bancos de Dados
SQL	–	Structured Query Language
SVN	–	Subversion – Controle de versão de código e de documentação
TADS	-	Tecnologia em Análise e Desenvolvimento de Sistemas
UFPR	–	Universidade Federal do Paraná
UML	–	Unified Modeling Language
PDO	–	PHP Data Object
IDE	–	Integrated Development Environment

## LISTA DE ILUSTRAÇÕES

FIGURA 1 DIAGRAMA DE CASOS DE USO.....	33
FIGURA 2 EXEMPLO DE ASSOCIAÇÃO.....	34
FIGURA 3 EXEMPLO DE AGREGAÇÃO.....	34
FIGURA 4 EXEMPLO DE COMPOSIÇÃO.....	35
FIGURA 5 EXEMPLO DE HERANÇA.....	35
FIGURA 6 EXEMPLO DE SEQUÊNCIA.....	37
FIGURA 7 DV1 - TELA INICIAL.....	55
FIGURA 8 DV2 - TELA DE LOGIN.....	56
FIGURA 9 - DV3 - TELA FORMULÁRIOS (DESLOGADO).....	56
FIGURA 10 DV3 - TELA FORMULÁRIOS (LOGADO).....	57
FIGURA 11 - DV4 - TELA CADASTRO DE UM NOVO FORMULÁRIO.....	58
FIGURA 12 DV4 - TELA EDIÇÃO DO FORMULÁRIO.....	58
FIGURA 13 DV5 TELA VISUALIZA FORMULÁRIO.....	59
FIGURA 14 DV4 CADASTRO DE NOVO/EDITA FORMULÁRIO.....	63
FIGURA 15 DV6 TELA ADICIONA CAMPO:PROPRIEDADES.....	64
FIGURA 16 DV1 TELA INICIAL.....	70
FIGURA 17 DV2 TELA DE LOGIN.....	70
FIGURA 18 DV1 TELA INICIAL.....	75
FIGURA 19 DV2 TELA DE LOGIN.....	76
FIGURA 20 DV8 TELA CADASTRO DE TIPO DE DADO.....	76
FIGURA 21 DV3: TELA FORMULÁRIOS (PREENCHER).....	80
FIGURA 22 DV9 TELA DO QUESTIONÁRIO (GERADO DINAMICAMENTE).....	80
FIGURA 23 DV2 TELA DE LOGIN.....	84
FIGURA 24 DV10 QUESTIONÁRIOS.....	87
FIGURA 25 DV11 RESPOSTAS DO QUESTIONÁRIOS.....	87
FIGURA 26 DIAGRAMA DE CASO DE USO.....	94
FIGURA 27 DIAGRAMA DE SEQUÊNCIA: INCLUIR FORMULÁRIO.....	95
FIGURA 28 DIAGRAMA DE SEQUÊNCIA: ALTERAR FORMULÁRIO.....	96
FIGURA 29 DIAGRAMA DE SEQUENCIA: INCLUIR CAMPO.....	97
FIGURA 30 DIAGRAMA DE SEQUÊNCIA: ALTERAR CAMPO.....	98
FIGURA 31 DIAGRAMA DE SEQUÊNCIA: EXCLUIR CAMPO.....	99
FIGURA 32 DIAGRAMA DE SEQUÊNCIA: GERAR RELATÓRIO.....	100
FIGURA 33 DIAGRAMA DE SEQUÊNCIA: INCLUIR TIPO DE CAMPO.....	101
FIGURA 34 DIAGRAMA DE SEQUÊNCIA ALTERAR TIPO DE CAMPO.....	102
FIGURA 35 DIAGRAMA DE SEQUÊNCIA: EXCLUIR TIPO DE CAMPO.....	103
FIGURA 36 DIAGRAMA DE SEQUÊNCIA: INCLUIR TIPO DE DADO.....	104
FIGURA 37 DIAGRAMA DE SEQUÊNCIA: ALTERAR TIPO DE DADO.....	105
FIGURA 38 DIAGRAMA DE SEQUÊNCIA: EXCLUIR TIPO DE DADO.....	106
FIGURA 39 DIAGRAMA DE SEQUÊNCIA: PREENCHER FORMULÁRIO.....	107
FIGURA 40 DIAGRAMA DE SEQUÊNCIA: LOGIN.....	108
FIGURA 41 DIAGRAMA DE CLASSES.....	109
FIGURA 42 DIAGRAMA ENTIDADE-RELACIONAMENTO.....	110

## 1. INTRODUÇÃO

### 1.1. INTRODUÇÃO

A informação tornou-se uma necessidade crescente para qualquer setor da atividade humana, e esta, aliada à informática através de sistemas informatizados, vem com a principal proposta de ajudar o homem a manipular suas informações e facilitar o acesso a elas, tornando-as assim mais seguras e confiáveis.

Atualmente levando-se em conta que o setor de TI vem expandindo-se dentro da UFPR juntamente com a demanda da informação, o fluxo de documentos e a padronização de procedimentos, a gestão desses recursos ainda é um fator de dificuldade, pois até o presente momento as soluções disponíveis eram compostas por formulários estáticos impressos com preenchimento de forma manuscrita.

Por isso buscou-se elaborar formulários que facilitassem o processo de cadastramento de dados, sejam estes oriundos do administrador do sistema ou dos usuários. A idéia proposta contemplou a construção de formulários Web disponibilizados na internet, de tal forma que o próprio fornecedor da informação realize a inserção dos dados ao invés de formulários de papéis, tirando a carga inicial dos funcionários da universidade, reduzindo custos com gasto de material e contribuindo assim com o meio ambiente.

O projeto consiste em desenvolver diversos tipos de formulários, disponibilizados aos alunos do curso de Tecnologia de Análise de Sistemas, onde o administrador do sistema possa confeccionar modelos de formulários de forma fácil e personalizada. A criação de formulários é livre, de acordo com as necessidades do curso. A intenção do sistema proposto, é a de oferecer aos tomadores de decisão mecanismos eficientes de gerenciamento com flexibilidade, segurança e com o grande diferencial de ser ligado à internet.

O que fundamenta o fato do desenvolvimento ser voltado para Web é a facilidade de acesso, tendo em vista a possibilidade de se entrar no sistema através de qualquer computador que esteja conectado ao “mundo” da internet. Criando um cenário onde temos a centralização da informação com a descentralização do acesso.

## 1.2. OBJETIVOS DO PROJETO

Com base em preceitos ergonômicos e dentro dos conceitos da Engenharia de Software, buscou-se a elaboração de formulários de fácil entendimento e com layout agradável, que não canse o usuário durante o preenchimento, bem como a automação necessária para realizar a entrada correta dos dados.

Compreendida a natureza real do problema, buscou-se possibilitar a efetivação de dois tipos de formulários: estáticos e dinâmicos. No estático temos a opção de inserir dados sólidos, como informações pertinentes a alunos e cadastros gerais. Já no formulário dinâmico, pode-se confeccionar formulários personalizados, no qual é possível determinar os campos e as formas de resposta esperada. O que facilita muito, pois o administrador do sistema é capaz de elaborar estes formulários, podendo livremente determinar quais questões o formulário deve possuir bem como o estilo de resposta, pode ser uma entrada de texto livre, ou uma opção dentro de algumas respostas padrões, como por exemplo uma caixa de seleção entre outras.

Uma grande vantagem do sistema é a sua portabilidade, uma vez que o *software* não se restringe a uma área de atuação específica, sendo sua implantação viável em qualquer setor da UFPR que necessite de gerenciamento das informações através da criação de formulários dinâmicos.

O sistema oferece para a organização, a oportunidade de ter toda sua estrutura funcional organizada e documentada de forma centralizada, obtendo o controle total da tramitação e acompanhamento dos formulários cadastrados. Com isso proporcionará um acompanhamento em tempo real, modelando assim a dinâmica do setor de acordo com suas necessidades.

### 1.2.1. OBJETIVO GERAL

Desenvolver um sistema informatizado para modelar e implementar formulários, de forma dinâmica e flexível, de forma a automatizar processos e prover soluções para facilitar o trabalho de quem gerencia o funcionamento do setor, dando maior confiabilidade no trabalho, minimizando erros e agilizando os processos de acordo com as necessidades do departamento de Tecnologia em Análise e Desenvolvimento de Sistemas.

### 1.2.2. OBJETIVOS ESPECÍFICOS

Cadastrar e gerenciar formulários pré-definidos pelo administrador do sistema, disponibilizar senha de acesso aos formulários somente a seus respectivos tutores e possibilitar o gerenciamento e acompanhamento dos questionários<sup>1</sup> já preenchidos através de relatórios

### 1.2.3. JUSTIFICATIVA

O departamento TADS da UFPR, até hoje compreendia de formulários estáticos de papel, preenchidos manualmente no próprio setor pelos alunos solicitantes. O que muitas vezes tornava-se inviável pois se houvesse alguma alteração ou inserção no formulário atual, acarretava uma série de adaptações, prejudicando a agilidade do processo. Assim como necessitava da presença física do aluno no setor.

---

<sup>1</sup> Questionários a priori são os formulários já preenchidos.

A idéia de se construir uma ferramenta eficaz de gerenciar formulários dinâmicos trará mais autonomia para a coordenação do setor, que poderão criar diversos modelos de formulários a serem preenchidos, com entrada de dados direcionada, facilitando assim o preenchimento dos mesmos e criando uma base de dados única que facilitará na criação e manutenção dos mesmos. Assim como, por seu desenvolvimento estar voltado para Web conferirá ao usuário a facilidade de acesso, tendo em vista a possibilidade de se entrar no sistema através de qualquer computador que esteja ligado à rede mundial.

Assim, deu-se início à modelagem e a implementação de uma ferramenta capaz de elaborar formulários dinâmicos, com os quais é possível construir formulários de forma personalizada, sem a necessidade de codificação ou alteração do sistema.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1. REVISÃO BIBLIOGRÁFICA**

Os documentos são a forma mais fácil e usual de se trocar informações, no interior de ambientes informatizados. Estes documentos podem ser compostos por formulários dinâmicos, onde as informações são atualizadas a cada acesso, e por formulários estáticos, onde as informações não mudam quando acessadas [BAL94, WEI92].

Não podemos deixar de citar a internet, pois é a área cujo avanço tecnológico está sendo bastante estimulado. Com a sofisticação de *Homepages*, tornou-se necessária uma melhor estruturação dos documentos, Hiperdocumentos e facilidades de navegação. Segundo Conklin [CON87], a idéia de facilitar a



navegação entre documentos originou a proposição de uma nova forma de estruturação da informação.

A novidade que este trabalho propõem está fundamentada em alguns sistemas já existentes na net. Entre eles o *Wuffo*<sup>2</sup>, que se trata de uma ferramenta on-line para a criação de formulários, onde o usuário pode criar seus formulários personalizados de maneira gráfica e com muitas opções de customização, como obrigatoriedade de campos, tamanho, etc. Após criar o formulário, o usuário poderá salvá-lo e utilizá-lo no web site. Outra ferramenta muito semelhante e que também nos influenciou no desenvolvimento do sistema é conhecido como pForm<sup>3</sup>. Da mesma maneira que o Wuffo, ele está disponível na internet e conta com a criação dos formulários de maneira prática, porém nenhuma dessas ferramentas são exclusivas para oferecer suporte ao gerenciamento das informações recolhidas através do formulário, elas simplesmente o geram dinamicamente para que sejam embutidos em outros sites. O sistema proposto neste trabalho, tem por objetivo não somente gerar dinamicamente o formulário como oferecer meios para que todos os dados oriundos deles sejam gerenciados.

#### 2.1.1. LINGUAGEM E AMBIENTE DE DESENVOLVIMENTO

Como o sistema proposto está voltado para o ambiente de desenvolvimento Web, ou seja, possui uma arquitetura baseada na Web. Portanto, do lado do servidor a tecnologia utilizada é a linguagem PHP e do lado do cliente é o HTML/JavaScript<sup>4</sup>.

---

<sup>2</sup> Maiores detalhes em [HTTP://www.wuffo.com](http://www.wuffo.com)

<sup>3</sup> Maiores detalhes em [HTTP://www.pform.com](http://www.pform.com)

<sup>4</sup> JavaScript é uma linguagem de Script que possibilita interagir com o conteúdo de uma página HTML. Maiores detalhes sobre esta linguagem podem ser encontrados em Goodman(2004).

Para o lado do servidor foi utilizado o sistema Apache<sup>5</sup>, que implementa um servidor Web, juntamente com o banco de dados MySQL, uma vez que é um SGBD robusto e versátil para uso conjunto com a linguagem PHP.

A respeito do PHP, pode-se dizer que é "(...)uma linguagem de criação de scripts do lado do servidor, que pode ser incorporada ao HTML ou utilizada como um binário independente."(CONVERSE,2003). Suas características principais são: código aberto e gratuito, é otimizada para trabalhar com o banco de dados MySQL e com o servidor Apache, sintaxe similar a linguagem C, possibilidade de programação imperativa e/ou orientada a objetos, além do mais é uma linguagem popular, muito utilizada nos dias de hoje.

Embora muito rica em seus recursos, a linguagem PHP é dita fracamente tipada. Para a implementação de grandes sistemas, o que torna-se muitas vezes um problema por se perder a garantia e a segurança da utilização das estruturas de dados.<sup>6</sup>E por ser uma linguagem interpretada, muitos erros sintáticos só são percebidos em tempo de execução. Apesar disso, a facilidade de se criar e trabalhar com estruturas de dados é muito útil quando se trata de sistemas Web.

### 2.1.2. ARQUITETURA DE SISTEMAS WEB

O Sistema Gerenciador de Formulários Eletrônicos foi modelado tendo por base o conceito de aplicação cliente/servidor e foi construído nas três camadas

---

<sup>5</sup> Maiores detalhes em <http://www.apache.org/>.

<sup>6</sup> Há muitas vantagens em se desenvolver através de uma linguagem fracamente tipada ,tem-se uma maior flexibilidade no tratamento dos dados.Muitos desenvolvedores defendem esta abordagem,contudo,ressaltam a necessidade de se tomar um maior cuidado durante a codificação do sistema.Sebesta(2005)traz em sua obra uma entrevista(pags:216 e 217)com o desenvolvedor Rasmus Lerdorf,responsável pela infra-estrutura do site Yahoo o qual descreve as vantagens de se utilizar linguagens de Script,dentre elas o PHP,para solucionar problemas.

seguintes: camada de apresentação, servidor de aplicação e servidor de banco de dados. Nesse sistema, o usuário faz uso de um navegador Web (Web Browser) como *front-end*<sup>7</sup> da aplicação, onde se conectando a uma rede na internet lhe proporcionará o acesso ao servidor de aplicação. O servidor de aplicação por seguinte armazena e executa as regras de negócio, acessando caso for necessário, um terceiro servidor que exerce a função de persistência de dados.

Em particular, para o sistema gerenciador de formulários Eletrônicos uma vasta parte da codificação foi escrita para proporcionar uma maior interação no lado do usuário.

Com base na obra de BENETT (1997), pode-se descrever que a arquitetura de sistemas Web é composta pelos seguintes itens:

- Um navegador (Web Browser) que fornece a interface visual com o usuário;
- Um servidor que contém a lógica da aplicação;
- O fluxo de dados sempre é iniciado pelo usuário no qual realiza requisições ao servidor e aguarda a resposta do mesmo.

Além disso, a arquitetura possui as seguintes características:

- Geralmente não há necessidade de se instalar qualquer aplicativo adicional na máquina do usuário;
- A sessão de trabalho, geralmente só existe durante o processamento da requisição do usuário pelo servidor; apesar de que existam recursos para se gravar por algum tempo os dados que venham a ser necessários existir em memória;

---

<sup>7</sup> Em projetos de software, front-end é parte do sistema que interage diretamente com o usuário

- O sistema é dito on-line, pois está sempre disponível desde que o usuário tenha acesso a um navegador e esteja conectado ao servidor de aplicação.

### 2.1.3. FORMULÁRIOS ESTÁTICOS E DINÂMICOS

Cury (2005) afirma que formulário é um documento com campos pré-impressos onde são preenchidos os dados e as informações, e que permite a formalização das comunicações, o registro e o controle das atividades das organizações.

E segundo Oliveira (2005) a atividade de organização e métodos é a que fornece os subsídios necessários para a elaboração e controle de formulários.

Sistemas Web apresentam grande facilidade para o desenvolvimento de formulários, habitualmente chamados de *Web Forms* ou simplesmente formulários HTML, segundo Crane (2007).

O uso de formulários dispensa o uso de papéis na coleta da informação e através de seus recursos computacionais disponíveis através de formulários HTML, propicia uma experiência positiva, pois aumenta a interatividade com o usuário.

A grande vantagem a que se propõe este projeto, é a forma com a qual pode ser elaborado, cuja implementação poderá ser elaborada de maneira estática, onde é inicialmente especificada e logo depois de codificada, ou poderá ser feita de forma dinâmica, o que possibilita ao administrador um grande poder de personalização.

Em suma, os campos de um formulário HTML, apresentam no mínimo, dois atributos: um identificador e um valor associado a este. Frequentemente, este identificador está mapeado a um elemento de um repositório. Um formulário estático

tem por principal perfil, manter constante esse mapeamento. Já nos formulários dinâmicos, o mapeamento deverá ser feito durante a carga do mesmo.

Tem-se como exemplo em formulário estático, uma entrada de texto mapeada para um campo qualquer de uma tabela de um banco de dados. Nesse caso, o principal trabalho da camada de aplicação é inserir o valor atribuído ao identificador de entrada de texto. Já, formulários dinâmicos apresentam um grau de complexidade maior, haja visto em que não há este mapeamento forte entre os campos do formulário e o modelo da camada de aplicação. Dessa maneira, torna-se necessário um modelo que realize de forma dinâmica a associação entre os campos do formulário e os campos do banco de dados.

Outros pormenores que não podemos deixar de citar, estão relacionados ao layout de visualização dos componentes, à limitação dos controles disponíveis, ao tratamento das informações inseridas e ao modelo de dados que tornam a implementação de formulários dinâmicos uma tarefa dispendiosa. Por outro lado, um sistema com gerenciamento de formulários dinâmicos pode proporcionar uma grande capacidade funcional. Uma vez que o administrador possa determinar seus próprios formulários, este passa a ter uma grande flexibilidade de personalização e independência de manutenção do sistema.

## 2.2. FUNCIONALIDADES E VANTAGENS

O sistema gerenciador de formulários eletrônicos deve ser capaz de fornecer a entrada de dados de cadastros básicos, a elaboração de formulários dinâmicos e uma forma de gerenciar as informações armazenadas.

Além destas funcionalidades, a ferramenta deve possuir um sistema de Login para controle de acesso, onde somente o administrador responsável pela criação de um determinado formulário, poderá posteriormente modificá-lo.

No sistema de *formulários dinâmicos eletrônicos* é necessário definir primeiramente os processos do mundo real para um modelo computacional e processável; após definido, o administrador do sistema poderá cadastrar os modelos de formulários e deixar a disposição dos usuários.

## 2.3. FUNCIONALIDADES GERAIS

Os modelos de formulários cadastrados serão visualizados através de qualquer navegador que irá se comunicar com o servidor Apache, onde os dados ficarão armazenados em um SGBD (Sistema de Gerenciamento de Banco de Dados) MySQL. Para modelagem do projeto será utilizado a UML, que permite que o projeto seja visualizado em diagramas padronizados com uma notação gráfica. Abaixo serão descritas as principais funcionalidades do sistema proposto, tanto do administrador do sistema, quanto do usuário.

### 2.3.1. ADMINISTRADOR DO SISTEMA

**Manter Formulário:** O administrador poderá editar e elaborar formulários específicos, incluindo propriedades referentes aos campos, de acordo com a necessidade de cada respectivo formulário. Assim como visualizá-los depois de prontos.

**Manter Campo:** O administrador poderá fazer a inclusão, alteração e exclusão de Campo(s) do Formulário.

Gerar relatório: O administrador poderá listar os questionários preenchidos, (em formato de planilha eletrônica)

Manter tipo de campo: O administrador poderá cadastrar os tipos de campos existentes que poderão ser utilizados na criação de novos formulários assim como editar, excluir, visualizar e listar os respectivos campos.

Manter tipo de dado: O administrador poderá cadastrar os tipos de dados existentes que poderão ser utilizados na criação de novos formulários assim como editar, excluir, visualizar e listar os tipos de dados.

### 2.3.2. USUÁRIO DO SISTEMA

Preencher Formulário: O usuário do sistema poderá escolher um determinado formulário e fazer seu o seu preenchimento.

## 3. METODOLOGIA

### 3.1. METODOLOGIA DO TRABALHO

#### 3.1.1. ORIENTAÇÃO A OBJETO

Através da orientação a objetos, podemos modelar o mundo real em classes e instâncias, onde cada classe é a estrutura de uma variável. Nela, podemos declarar atributos e métodos que poderão ser executados ou acessados nas instâncias da mesma classe. As classes possuem uma função muito importante na modelagem orientada a objetos: através delas é possível dividir o problema, modularizar a aplicação e reduzir o nível de acoplamento do software.

Devido ao alto nível de abstração que se pode obter, o modelo orientado a objeto pode ser utilizado desde o início da concepção do software até seu

desenvolvimento, possibilitando assim facilidade em se relacionar as partes do sistema aos conceitos que ele implementa, e a implementação com as decisões do projeto. Isso possibilita uma boa manutenção do software, durante e após o desenvolvimento, porque os dados e as funções que pertencem ao mesmo conceito ficam agrupados.

A orientação a objeto prioriza a estrutura de um sistema às suas funções, essa característica é importante porque as funções que um sistema realiza podem sempre mudar com o tempo, enquanto raramente ocorrem mudanças na estrutura

### 3.1.2. UML – UNIFIED MODELING LANGUAGE

UML (*Unified Modeling Language*, ou linguagem de modelagem unificada) é a padronização das metodologias de desenvolvimento de sistemas baseados na orientação a objeto. Dá suporte ao desenvolvimento de sistemas de software de grande porte através de métodos de análise e projeto, que modelam esse sistema de modo a fornecer a todos os envolvidos uma compreensão única do projeto.

Com a escolha da abordagem orientada a objetos, a utilização da UML traz a possibilidade de documentação adequada das informações sobre o sistema, visualização da estrutura do software através de representação gráfica, especificação desde a fase de análise até a fase de implementação e testes do sistema, realização de mapeamento dos modelos gerados.

“A UML não é um método, é um modelo de linguagem que é utilizada para visualizar, especificar construir e documentar artefatos de software” (SPÍNDOLA, p.18). O modelo de linguagem é a notação que o método usa para descrever o



projeto. O processo são os passos que devem ser seguidos para se construir o projeto.

Com ela podemos fazer uma modelagem visual de maneira que os relacionamentos entre os componentes do sistema sejam mais bem visualizados, compreendidos e documentados.

“A UML é baseada em Diagramas: ênfase visual, onde vários aspectos fundamentais na modelagem de sistemas são abordados” (SILVA , 2004), tais como:

- Funcional: estrutura estática e interação dinâmica,
- Não funcional: tempo de processamento, confiabilidade e produção,
- Organizacional: organização do trabalho, mapeamento e código.

### 3.1.3. CARACTERÍSTICAS DA UML

A UML é uma linguagem de modelagem, não uma metodologia para desenvolvimento de sistemas. “Absorveu influências de outras técnicas de modelagem” (SILVA, 2004):

Benefícios:

- Realça os relacionamentos entre os depósitos de dados de um DFD que de outro modo só seriam percebidos nas especificações de processos.
- Facilita descobrir funções necessárias no DFD.
- Organiza a estrutura de dados ajudando a evitar a redundância de dados.

O objetivo da UML é descrever qualquer tipo de sistema em termos de diagramas orientados a objetos. Ela é utilizada para a visualização do problema pela

equipe de desenvolvimento de software. Segundo Spíndola (p. 18) “evita a perda de informação durante a comunicação entre a equipe. Facilita a construção do software porque prevê uma notação comum para o entendimento compartilhado sobre o software que se está construindo”.

“A UML permite a construção de modelos precisos, não ambíguos e completos para a especificação do software que será construído” (BOOCH, RUMBAUGH, JACOBSON, 2000, p.16).

Os modelos construídos usando a UML têm uma dimensão de “projeto” que é independente de uma linguagem de programação e permitem a tradução entre os modelos e códigos dessas linguagens. Podem ser implementados em qualquer linguagem de programação ou permite também a Engenharia Reversa na qual os modelos são gerados a partir do código-fonte.

Todo projeto de software envolve muita documentação desde a fase inicial até o produto final. Os modelos criados durante o desenvolvimento fazem parte da documentação do software.

“Arquitetura de Sistema de Software é um conjunto de decisões significativas sobre a organização do sistema de software, a seleção de elementos estruturais e suas interfaces, o comportamento do sistema, composição dos elementos estruturais e comportamentais” (BOOCH, RUMBAUGH, JACOBSON, 2000, p. 32).

A arquitetura de um sistema apresenta diferentes visões do sistema que está sendo modelados, diferentes pontos de vista devem ser usados para refletir os aspectos desejados. Cada visão mostra um conjunto de aspectos do sistema numa notação adequada à sua compreensão.

### 3.1.4. APLICAÇÕES DA UML

Pode ser utilizada nos seguintes sistemas:

**Sistemas de Informação:** Pesquisa, edita, armazena e mostra informações aos usuários com grande quantidade de dados que são armazenado em banco de dados relacionais e orientado a objetos.

**Sistemas Técnicos:** Controlam e mantém equipamentos de telecomunicação, militares e de processos industriais possuindo uma interface especial e ao contrario dos Sistemas de Informação possui menos programação sendo assim basicamente um sistema Real-Time.

**Sistemas real-time integrados:** São executados em hardware, possuem programação de baixo nível e suporte real-time.

**Sistemas distribuídos:** Distribuídos em máquinas onde os dados são transferidos de forma simples entre as maquinas. Os sistemas de mecanismos são sincronizados onde é mantida a integridade dos dados.

**Sistemas de software:** Onde são definidas as estruturas que são utilizadas por outros softwares. Sistemas Operacionais, Banco de Dados e ações de baixo nível no hardware.

**Sistemas de negócios:** Os processos de negócios descrevem seus objetivos e especificações dentro das regras (leis, estratégias de negócios etc.), do atual trabalho desempenhado.

É importante saber que os sistemas não possuem uma ou mais características apresentadas acima, mas sim varias características ao mesmo tempo.

Um exemplo disso seria o Sistema de Informações atual que possuem características distribuídas (Real-Time), onde a UML suporta todos os tipos de Sistemas.

#### 3.1.4.1. FASE DO DESENVOLVIMENTO DE UM SISTEMA EM UML

O desenvolvimento de um sistema de software envolve diversas fases onde, em cada uma delas, diversas atividades são desenvolvidas utilizando a UML. As fases de desenvolvimento são: Análise de requisitos, fase de análise, fase de projeto, fase de programação e fase de testes.

#### 3.1.4.2. ANÁLISE DE REQUISITOS

A análise de requisitos é fundamental para o desenvolvimento de sistemas, pois trata justamente de descobrir o que o cliente quer com o sistema. A análise de requisitos está associada ao processo de descobrir quais são as operações que o sistema deve realizar e quais são as restrições que existem sobre estas operações (WAZLAWICK, 2004).

Nessa etapa são coletadas as informações sobre os dados de interesse da aplicação. Nessa etapa são realizadas entrevistas com os futuros usuários do sistema, análise de documentações fazendo uma descrição dos requisitos da aplicação mais detalhada e clara possível.

Os principais requisitos do sistema são capturados através do desenvolvimento dos diagramas de caso de uso do sistema. Os requisitos são classificados em Funcionais e Não-funcionais. O primeiro requisito define o que o sistema deve fazer, e o segundo define as restrições sobre como o sistema deve desempenhar suas funções.

“É Importante salientar que os requisitos do sistema são informações que o cliente ou usuário solicitam, e não informações planejadas pelo analista” (WAZLAWICK, 2004).

#### 3.1.4.3. ANÁLISE

É a fase de análise dos requisitos obtidos, nessa etapa é realizada a modelagem conceitual que descreve a informação que o sistema vai gerenciar. “A análise de domínio está relacionada à descoberta das informações que são gerenciadas no sistema, ou seja, à representação e transformação da informação modelando-as em classes e objetos” (WAZLAWICK, 2004).

“O modelo funcional pode ser representado através dos contratos de operações de sistema” (WAZLAWICK, 2004).

O Modelo Conceitual ou Modelo de domínio ilustra as classes conceituais do sistema. Analisa o domínio do problema e não do domínio da solução. Não deve ser confundido com o modelo de dados.

O Modelo Conceitual representa classes conceituais ou objetos do mundo real no domínio de um problema, ou seja, nessa fase são modeladas as principais classes do sistema que estão presentes no domínio do sistema deixando para a fase de projeto a modelagem de classes técnicas que gerenciem banco de dados, interfaces, comunicações.

#### 3.1.4.4. PROJETO

A fase de projeto enfatiza a proposta de uma solução que atenda os requisitos da análise. “A fase de análise é uma investigação para tentar descobrir o

que o cliente quer, o projeto consiste em propor uma solução com base no conhecimento adquirido na análise” (WAZLAWICK, 2004).

Na fase de projeto podem-se ter diagramas mais técnicos e mais precisos:

- Diagramas de classes a partir de uma perspectiva de software.
- Diagramas de seqüência para cenários comuns, utilizando os cenários mais importantes dos casos de uso e fazendo diagramas de seqüência para descobrir como acontecerá no software.
- Diagramas de pacotes para mostrar a organização em larga escala do software.
- Diagramas de estados para as classes com históricos de vida complexos.
- Diagramas de distribuição para mostrar o layout físico do software.

Na fase de Projeto espera-se que a implementação do código acompanhe os diagramas. Essa fase resulta no detalhamento das especificações para a fase de programação do sistema.

#### 3.1.4.5. IMPLEMENTAÇÃO (PROGRAMAÇÃO)

A utilização de técnicas sistemáticas nas fases de análise e projeto faz com que o processo de geração de código possa ser automatizado. Neste caso, cabe ao programador dominar as características específicas das linguagens, ferramentas, frameworks e estruturas de dados para adaptar o código gerado aos requisitos indicados quando necessário (WAZLAWICK, 2004).

A Fase de programação é separada das outras fases, pois é nessa fase que os modelos criados e analisados serão convertidos em código, para implementação do software.

#### 3.1.4.6.      TESTE

A fase de testes envolve os testes de unidade, feitos pelo programador, para verificar se os componentes gerados atendem à especificação feita nas fases anteriores, e aos testes de caso de uso, que visam verificar a adequação do sistema aos requisitos inicialmente levantados (WAZLAWICK, 2004).

Os objetivos da fase de teste são:

- Verificar se as especificações da etapa de projeto refletem exatamente os requisitos levantados na fase de análise de requisitos e as definições na fase de análise, ou seja, verificar se o sistema funciona conforme os casos de uso especificados e se as classes implementadas estão se relacionando e cooperando entre si para garantir que o sistema gerado esteja de acordo com o sistema desejado.

- Verificar se não existem erros de lógica no projeto e no código, no fluxo de dados, no entendimento de requisitos, de codificação, de interface em todas as fases do projeto. Nesta fase devemos identificar e corrigir os erros.

O sucesso da fase de testes consiste em descobrir os erros e corrigi-los, para que o usuário final teste e verifique que o sistema está de acordo com o desejado.

#### 3.1.4.7.      DIAGRAMA DE CASOS USO

Definição: O Diagrama de Caso de Uso dá uma visão geral de todas as funções do sistema mostrando a interação entre atores e Casos de Uso e mesmo entre os Casos de Uso.

Elementos:

**Ator** é alguém ou alguma coisa que interage com o sistema, é quem ou o que usa o sistema. O ator representa um papel, não um usuário individual do sistema. Atores podem ser humanos ou sistemas automatizados.

**Caso de Uso** representa uma funcionalidade completa do sistema, conforme percebida por um ator. É um conjunto de seqüências de ações que um sistema executa, gerando um resultado observável que interessa a determinado ator.

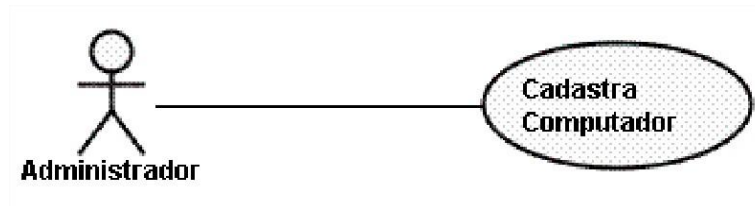
**Relacionamentos** são divididos em três tipos:

Generalização ou Herança: Usado quando um Caso de Uso é semelhante a outro, mas “faz um pouco mais”. Neste caso, o segundo caso é uma especialização do primeiro.

Uso (Include): Ocorre quando existe uma parte do comportamento de um caso de uso que pode ser usado em outro caso, ou seja, uma rotina comum.

Extensão (Extend): Está sujeita a uma determinada condição, ou seja, o comportamento do caso de uso estendido é inserido somente se uma condição for Verdadeira.





**Figura 1 Diagrama de casos de uso**

O Diagrama de Caso de Uso fornece um modo fácil de descrever a visão externa do sistema e suas interações com o mundo exterior, representando uma visão de alto nível da funcionalidade do sistema mediante uma requisição do usuário. Ele é bastante importante porque vai ser a base do processo de desenvolvimento do sistema.

#### 3.1.4.8. DIAGRAMA DE CLASSES

“O Diagrama de Classes é o mais utilizado e mais importante da UML. Serve de apoio para a maioria dos demais diagramas.” (GUEDES, 2008)

Ele exemplifica a estrutura das classes usadas pelo sistema, determinando seus atributos, métodos e relacionamentos. Um diagrama de classes é um gráfico que mostra a estrutura do sistema, mostra as classes e os relacionamentos entre elas, descreve os aspectos estáticos do sistema e não mostra informações temporais.

Uma classe é um molde para a construção de objetos que define os atributos e operações comuns aos objetos. Uma classe tem nome, uma lista de atributos, comportamentos (operações) e relacionamentos com outras classes. Um objeto é uma instância de uma classe. “Os atributos são propriedades do objeto e as operações são ações que o objeto é capaz de efetuar” (SPÍNDOLA, p.20),

normalmente as operações que um objeto pode fazer alteram os dados ou o estado do objeto.

Os relacionamentos entre as classes podem ser:

- **Associação:** “representa uma relação estrutural entre duas classes indicando que duas classes se comunicam” (SPÍNDOLA, p.21) É representada através de uma linha simples com um nome, geralmente identificados por verbos na especificação do problema.

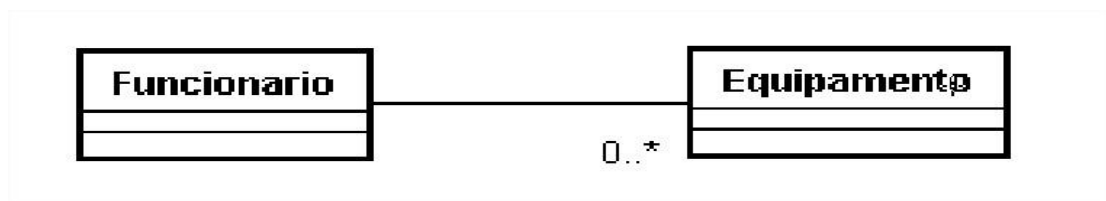


Figura 2 Exemplo de Associação

- **Agregação:** É um tipo especial de associação onde o elemento associado corresponde a uma parte do elemento principal. As partes só existem se o todo existir e vice-versa.

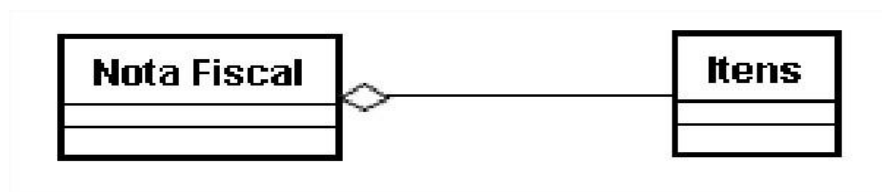


Figura 3 Exemplo de Agregação

- **Composição:** “é uma associação que expressa a semântica „parte de” mais fortemente. Neste tipo de relacionamento se o todo for destruído, suas partes

também serão” (SPÍNDOLA,p.21). É representada através de uma linha que conecta a classe agregada com a outra classe através de um losango preenchido ao lado da classe agregada. Esse relacionamento é uma forma restrita de agregação.

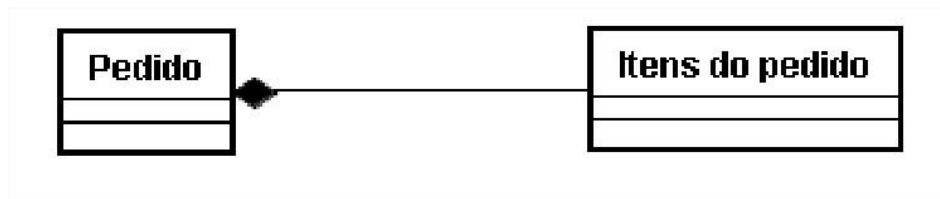


Figura 4 Exemplo de composição

- **Generalização ou Herança:** “relaciona classes generalizadas com suas especializações” (SPÍNDOLA, p.21). A classe generalizada é representada por um triângulo unido a classe e as especializações são ligadas a esse triângulo por linhas simples.



Figura 5 Exemplo de herança

#### 3.1.4.9. DIAGRAMA DE SEQUÊNCIA

O Diagrama de Seqüência demonstra a ordem cronológica que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Ele baseia-se no Caso de Uso e no Diagrama de Classes para determinar os objetos

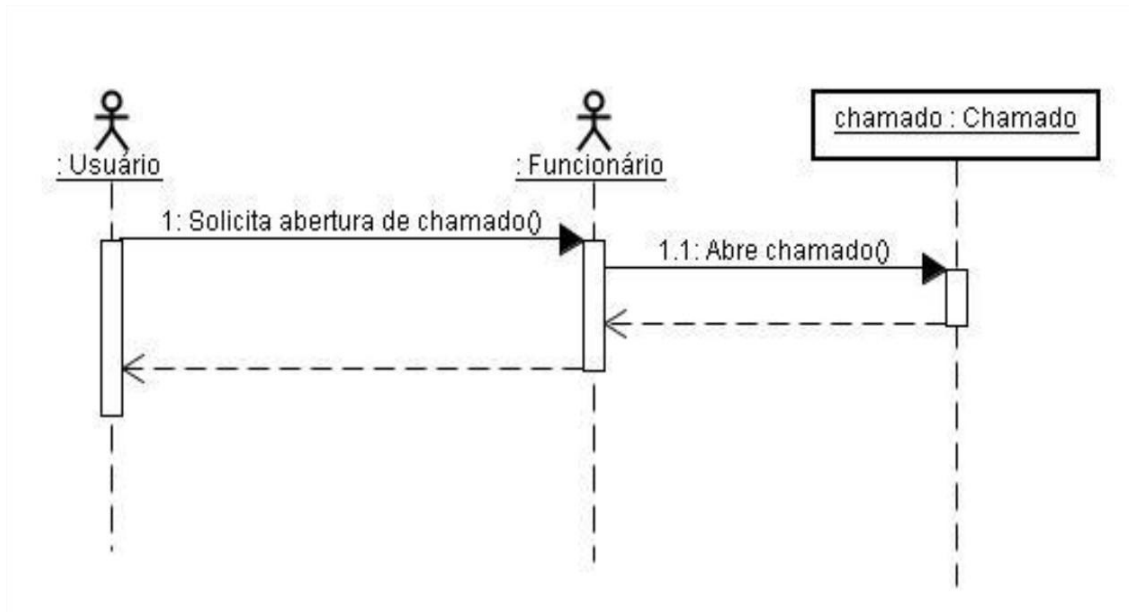
envolvidos em um processo. Ele também identifica o evento ou ator gerador do processo modelado.

Um Diagrama de Seqüência é um diagrama que mostra, para o cenário específico de um caso de uso, os eventos que os atores externos geram sua ordem e os eventos entre sistemas.

Todos os sistemas são tratados como uma caixa-preta, a ênfase do diagrama está nos eventos que atravessam a fronteira do sistema entre atores e outros sistemas.

Deve ser feito o Diagrama de Seqüência para a seqüência de sucesso principal do caso de uso e para cenários freqüentes ou alternativas complexas. “Um Diagrama de Seqüência mostra eventos de sistema para uma seqüência, ou cenário; portanto, é gerado a partir da inspeção de um caso de uso” (LARMAN, 2004, p.144).

Os termos mostrados nos Diagramas de Seqüência são bem resumidos e podem necessitar de uma explicação adequada, de modo que, durante o trabalho de projeto fique bem claro o que está entrando e saindo do sistema. Se isto não foi explicado nos casos de uso, então o glossário pode ser usado, no entanto, deve sempre haver um uso significativo do glossário em caso de decisão. Caso contrário, simplesmente teremos um trabalho desnecessário e que não acrescenta valor.



**Figura 6 Exemplo de sequência**

### 3.1.5. ARQUITETURA EM CAMADAS

Camadas são divisões lógicas de componentes agrupados por responsabilidades em comum. Isso significa dizer que uma camada irá agrupar classes, pacotes e outros componentes que possuam responsabilidades em comum. As classes responsáveis por uma determinada funcionalidade serão agrupadas em uma camada. Essa separação promove a coesão na medida em que classes relacionadas estão juntas, e evita o acoplamento porque a comunicação entre camadas é feita de maneira controlada.

Idealmente, camadas se comunicam apenas com camadas adjacentes. Mas nem sempre este princípio pode ser seguido, porém o ideal é que as camadas se comuniquem apenas com as camadas vizinhas.

Nos sistemas modernos, existem dois tipos de Camadas: Físicas e Lógicas. Normalmente as Camadas Físicas são chamadas em inglês de *tiers* enquanto as lógicas são conhecidas como *layers*. Isso não é uma regra e em português usa-se a mesma palavra para ambas. Não existe relação predefinida entre Camadas Lógicas e Físicas num sistema, uma Camada Física pode hospedar diversas Camadas Lógicas ou uma Camada Lógica pode se espalhar por diversas Camadas Físicas.

Como qualquer técnica, o uso de Camadas tem vantagens e desvantagens. Algumas das vantagens das Camadas:

Reduzem complexidade: agrupam componentes e simplificam a comunicação entre eles;

Reduzem dependência/acoplamento: a regra de comunicação evita dependências diretas entre componentes de camadas diferentes;

Favorecem a coesão: componentes de responsabilidades relacionadas são agrupados;

Promovem reusabilidade: camadas podem ser reutilizadas em outros sistemas ou podem ser substituídas;

É um Padrão Arquitetural conhecido: facilita a comunicação e entendimento entre desenvolvedores.

E entre as desvantagens: Limitadas pela tecnologia: algumas regras precisam ser quebradas por limitações tecnológicas;

Podem complicar um sistema muito simples: não é qualquer sistema que exige o uso de camadas;

Possibilidade de “overdose”: muitos arquitetos acabam criando camadas demais e tornando a aplicação extremamente complexa.

### 3.1.6. DESIGN PATTERNS

Um *pattern* (padrão) descreve uma solução já testada para um problema recorrente em um projeto, dando ênfase especial ao contexto do problema e às conseqüências e impacto da solução. Segundo GAMMA, HELM, JOHNSON e VLISSIDES (1995), *patterns* são a descrição de classes e objetos os quais se comunicam e que são customizados para resolver um problema comum de projeto em um contexto específico. *Design pattern* pode então ser definido como uma solução para um problema em um determinado contexto. Em termos de orientação a objetos, *design patterns* identificam classes, instâncias, seus papéis, colaborações e distribuição de responsabilidades.

A utilização de *design patterns* é importante porque esses padrões já foram testados; eles refletem a experiência, o conhecimento adquirido e as idéias de desenvolvedores que tiveram sucesso utilizando-os em seus próprios trabalhos. *Design patterns* são reutilizáveis, pois provêm uma solução pronta que pode ser adaptada a diferentes problemas, conforme necessário. São significativos, proporcionam um vocabulário usual de soluções que podem expressar grandes soluções de forma sucinta (PATTERNS, 2006).

Sua utilização força uma forma otimizada e clara de comunicação entre os desenvolvedores, documentação e maiores possibilidades de exploração para alternativas de soluções para o projeto. Melhora a qualidade geral do programa, pois reduz a complexidade do código oferecendo uma abstração das classes e instâncias e proporciona redução no tempo de aprendizado de novas bibliotecas ou funções.

### 3.1.7. MVC - MODEL VIEW CONTROLLER

A utilização da arquitetura de aplicativos em três camadas permite a separação das camadas de dados, controle e visualização. Oferece vantagens como à otimização das habilidades de equipes, a redução de custos associados ao desenvolvimento, além de favorecer a extensão e reutilização do código.

A separação do sistema em camadas possibilita maior facilidade e clareza na implementação da persistência de dados, do controle de segurança, da comunicação em rede e do fluxo de visualização. Essa separação permite que os sistemas possam ser mais facilmente modificados e estendidos para atender a novas exigências, bem como possibilita que a interface com o usuário apresente várias visões de um só modelo, sem interferir com a lógica de negócio.

O *design pattern* MVC (ModelViewController, ou modelovisãocontrolador) Permite a separação do modelo de dados das várias formas que o dado pode ser acessado e manipulado. Um sistema MVC é dividido em modelo de dados, conjunto de visões e conjunto de controladores. MVC é útil principalmente para aplicações grandes e distribuídas, em que dados idênticos são visualizados e manipulados de formas variadas. A chave para o MVC é a separação de responsabilidades: as visões fornecem a interface do usuário, os controladores são responsáveis por selecionar uma visão apropriada e por fazer alterações no modelo de dados, e o modelo é responsável por representar os dados base da aplicação.

### 3.1.8. APLICAÇÃO DA METODOLOGIA “METADADOS”

Metadados são freqüentemente definidos como dados sobre dados, entretanto há outras definições dependendo do uso. Na maioria dos sistemas



computacionais, metadados têm essencial importância ao darem significado para os dados armazenados. Atualmente, o interesse na pesquisa de metadados cresce em razão da necessidade dos sistemas de informação oferecerem maior facilidade de manutenção e flexibilidade. Usando um modelo de dados baseado em metadados, as aplicações podem se tornar mais dinâmicas, além de "economizarem" tabelas e classes.

Para que o sistema pudesse oferecer recursos como a geração dinâmica de formulários e a coleta de dados por meio deles, foi necessário utilizar uma modelagem que oferecesse flexibilidade. No sistema, objetivo deste projeto, teve que se pensar em metadados para que o administrador tivesse como criar os formulários, configurando os campos existentes neles, os tipos de dados permitidos e comportamentos que refletirão em requisitos no momento do preenchimento do questionário, deste modo tudo que se referir à configuração de formulários poderá ser tratado como metadados.

### 3.2. TECNOLOGIA E SOFTWARE

Segue a lista e descrição das ferramentas utilizadas no desenvolvimento do sistema.

#### 3.2.1. ESPECIFICAÇÃO E MODELAGEM UML DO SISTEMA

##### 3.2.1.1. JUDE

Versão: Community 5.3

JUDE (Java and UML *Developer Environment*) é uma poderosa ferramenta gráfica para modelagem UML. Suporta modelagem de softwares orientados a objeto desenvolvidos em Java. Possui características como adição de métodos no

diagrama de seqüência e a reflexão dessa alteração no diagrama de classes. Apresenta boa performance.

### 3.2.2. LINGUAGEM E AMBIENTE DE DESENVOLVIMENTO DO SISTEMA

#### 3.2.2.1. PHP

Versão: 5.3

PHP (*Hypertext Preprocessor*): é uma linguagem de programação interpretada, livre e muito utilizada para gerar conteúdo dinâmico na *World Wide Web*. Adotamos o PHP 5 uma vez que terá compatibilidade com o site do MOODLE(do curso TADS) que também é desenvolvido em PHP.

#### 3.2.2.2. NETBEANS

Versão: 6.9.1

O Netbeans é um ambiente de desenvolvimento integrado, gratuito e código aberto para desenvolvedores de software na linguagem Java, C/C++, PHP e outras. O IDE é executado em muitas plataformas, como Windows e Linux.

#### 3.2.2.3. MYSQL

Versão: 5.1.4

4.2.1 MySQL: é um banco de dados completo, robusto e extremamente rápido. Como foi dado como opção pelo cliente, optamos pelo MySQL e também por maior conhecimento nesse banco de dados.

### 3.2.3. CAMADA DE APRESENTAÇÃO DO SISTEMA

#### 3.2.3.1. HTML

Especificação: 4.01

HTML (*HyperText Markup Language*, ou linguagem de formatação de hipertexto) trata-se de uma linguagem de marcação utilizada para produzir páginas na internet. Documentos HTML são feitos para prover estrutura lógica da informação destinada à apresentação de páginas da rede mundial de computadores.

#### 3.2.3.2. CSS

Versão: 2.1

CSS (*Cascading Style Sheet*, ou folha de estilo em cascata) é um mecanismo simples para adicionar estilos (fontes, cores, espaçamentos) aos documentos Web. O uso de CSS permite separar a marcação HTML da apresentação do site, ou seja, o HTML destina-se unicamente a estruturar e marcar o conteúdo, enquanto a CSS fica responsável pela parte visual do documento.

#### 3.2.3.3. JQUERY

Versão: 1.4.4

jQuery é uma excelente e leve biblioteca para desenvolvimento rápido de javascripts que interagem com uma página html, com ele podemos atribuir eventos, definir efeitos, alterar ou criar elementos na página, dentre diversas outras infinitudes de ações. Uma das principais vantagens do jQuery, é o fato de podermos escrever os scripts sem nenhuma inserção na página, exceto o link do arquivo javascript.

#### 3.2.3.4. JAVASCRIPT

Inicialmente conhecida por *LiveScript*, desenvolveu-se no Netscape para tornar seu browser mais poderoso e funcional, proporcionando a seus usuários uma experiência mais rica na web. Atualmente todos os browsers suportam JavaScript, apesar de haver diferenças de sintaxe entre um browser e outro. É uma linguagem utilizada principalmente para auxílio de desenvolvimento de páginas para a internet. É utilizada na criação de campos, validações e ações do formulário.

### 3.2.3.5. ASYNCHRONOUS JAVASCRIPT AND XML – AJAX

AJAX é uma técnica e não uma tecnologia, que reúne tecnologias conhecidas há bastante tempo, porém utilizadas separadamente. Ele propõe a junção dessas tecnologias (JavaScript, XML, DOM e CSS) para tornar a experiência do usuário mais rica e fascinante.

Uma aplicação Ajax elimina o natural '*reload*' de página na internet, introduzindo um intermediário, ou seja, uma ferramenta Ajax entre o usuário e o servidor. Em vez de carregar uma página, o browser carrega uma ferramenta Ajax, escrita em JavaScript. Esta ferramenta é responsável por renderizar a interface que o usuário vê e pela comunicação com o servidor, a ferramenta Ajax permite que a interação do usuário com a aplicação aconteça assincronamente, independente da comunicação com o servidor. Então o usuário nunca encarará uma janela branca do browser e um ícone de ampulheta, esperando pelo servidor para retornar algo.

Toda ação do usuário normalmente geraria uma requisição HTTP, ao invés disso, a forma de uma chamada JavaScript para a ferramenta Ajax. Qualquer resposta para a ação do usuário que não requer uma chamada ao servidor, como validações simples de dados, manipulação de dados na memória, e até mesmo alguma navegação é tratada pela própria ferramenta. Se a ferramenta precisa de

algo do servidor para responder, dados são submetidos para processamento, carregamento de código de interfaces adicionais, ou retorno de novos dados, a ferramenta faz essas requisições assincronamente, geralmente usando XML, sem parar a interação do usuário com a aplicação.

#### 3.2.4. CAMADA DE CONTROLE DO SISTEMA

Através da principal linguagem de programação utilizada neste projeto, que pode ser vista no item 3.2.2.1, foi implementado a camada de controle do sistema, onde classes receberão as requisições do usuário e executarão a lógica adequada sobre as classes da camada de modelo e persistência.

#### 3.2.5. CAMADA DE MODELO E DE PERSISTÊNCIA DO SISTEMA

Para a camada de modelo e persistência, foi utilizado, além da própria linguagem de programação já descrita no item 3.2.2.1, a extensão PDO que oferece uma interface para acesso ao banco de dados. Utilizando esta extensão, é possível criar uma abstração na camada de dados o que torna o sistema independente do tipo de banco de dados, ou seja, pode-se usar um banco de dados MySQL ou PostgreSQL apenas modificando o driver de acesso.

#### 3.2.6. REESCRITA DE URL

Utilizando as boas práticas de desenvolvimento para sistemas web, optou-se por utilizar a reescrita de URL que torna as URL's do sistema mais amigáveis para o usuário, além de ser possível esconder detalhes de qual tecnologia foi utilizada para o desenvolvimento como, por exemplo, a extensão da página, porém o mais interessante deste recurso está em poder escrever regras de reescrita que tornam a

URL uma descrição para um determinado recurso do sistema. Para conseguir utilizar tal recurso, é necessário utilizar o módulo `mod_rewrite` do servidor Apache.

### 3.2.7. FERRAMENTAS DE INFRAESTRUTURA DO SISTEMA

#### 3.2.7.1. APACHE

Versão: 2.2

Apache: é o mais bem sucedido servidor Web livre. É também compatível com o protocolo HTTP versão 1.1. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos, utilizando a API do software.

#### 3.2.7.2. SUBVERSION

Versão: 1.6

Subversion (também conhecido por SVN) é um sistema de controle de versão desenhado especificamente para ser um substituto moderno do CVS, que se considera ter algumas limitações.

#### 3.2.7.3. UNFUDDLE

Versão:

Unfuddle é um site que auxilia no gerenciamento de projeto de software e fornece mecanismos de controle de versão como o SVN descrito no item 3.2.7.2. Através dele é possível acompanhar o projeto seguindo as metas estipuladas.

### 3.3. DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento do projeto de formulários dinâmicos, foi usada a modelagem Orientada a Objetos, compreendendo o diagrama de casos de uso com a sua devida especificação, o diagrama de Classes e Seqüência.

Para o armazenamento dos dados, foi feito uso do servidor de banco de dados MySQL. Para disponibilizar o acesso ao sistema pelo navegador web, foi escolhido o servidor Apache o qual também é utilizado como servidor para aplicações feitas na linguagem de programação PHP.

Dados os requisitos do sistema, o curto prazo e as poucas reuniões com o orientador foi possível modelar uma primeira versão que serviu como base para a construção do sistema final. Utilizando materiais de pesquisa foi possível encontrar uma estrutura para iniciar o projeto de forma que propiciasse um rápido desenvolvimento, porém sem esquecer das boas práticas e padrões que pautam um projeto de software orientado a objetos.

Encontramos ferramentas que auxiliaram na construção do sistema, facilitando a implementação das funcionalidades que foram solicitadas. No primeiro momento após o desenvolvimento do protótipo e sua apresentação, constatou-se que alguns recursos poderiam ser melhorados afim de proporcionar uma melhor experiência para o usuário, isso norteou as implementações posteriores.

Mesmo com a escassez de tempo e ainda a complexidade do projeto, pode-se implementar de maneira eficaz as funcionalidades do projeto proposto. Visto que não há muita bibliografia específica sobre o assunto em português, todos os aspectos referentes a documentação, tiveram que tiradas da nossa própria experiência durante o desenvolvimento.

Dentro das expectativas e dos requisitos levantados ao longo deste processo, o resultado final obtido foi assumido como sendo muito bom e satisfatório.

### 3.3.1. ESTUDO DE CASO – GERENCIADOR DE FORMULÁRIOS ELETRÔNICOS

Contexto: Universidade Federal do Paraná.

Solicitante: TADS - setor do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Solicitação: Sistema para criação de formulários dinâmicos.

Sistema Atual: Não é informatizado, é composto por formulários impressos, preenchidos manualmente.

Problemas: Dificulta a agilidade do processo e na dinâmica dos cadastros.

Descrição do Processo: Foi determinada pelo cliente/orientador.

Requisitos:

- O sistema deve listar todos os formulários cadastrados pelo administrador.
- O administrador deverá poder cadastrar novos formulários, assim como editá-los e excluí-los.
- O administrador deverá poder visualizar o formulário depois de pronto.
- O administrador deverá poder gerar relatório dos questionários.

## 4. APRESENTAÇÃO DO SOFTWARE



#### 4.1. DIAGRAMA DE CASO DE USO

Conforme demonstrado no apêndice A.

#### 4.2. ESPECIFICAÇÃO DOS CASOS DE USO

##### 4.2.1. CASO DE USO: MANTER FORMULÁRIO

**Ator:** Administrador

**Visão geral:** Este caso de uso permite ao administrador do sistema o cadastro de novo formulário, a edição dos formulários, assim como sua visualização.

**Estado:** Especificado

**Pré-Condições:**

1. Usuário deve possuir o perfil de administrador;
2. Usuário deve estar logado no sistema.






**Cenário Principal:**

- O usuário abre a tela *“inicial” do sistema(DV1)* e depois de se logar na interface(DV2) poderá selecionar o menu “Formulário”.
- O sistema abre a tela (DV3) *“Formulários”* que contém *templates default de formulários existentes*.
- Em (DV3) na barra de menu principal quando Status do usuário for “Logado” aparecerá no menu os links: “Tipo de dado” e “Tipo de campo” onde o usuário terá acesso às configurações do sistema,*além do ícone “Novo”*.
- O usuário pressiona o ícone “Novo”.
- O sistema redireciona para a tela (DV4) “Cadastro de um Novo Formulário” onde o usuário primeiramente preenche os campos relativos as propriedades do formulário,tal como nome e descrição.E ao pressionar o botão submeter, o formulário será cadastrado.

- Na mesma interface os dados serão exibidos, referentes ao nome e descrição do formulário, assim como abrirá uma tabela denominada "Propriedades do Formulário e dos Campos", que contém um botão "Adicionar campos".
- Quando o usuário pressionar o ícone "Adicionar campos" abrirá uma janela denominada (DV6) "Adicionar campo:propriedades", onde o usuário digitará os dados pertinentes ao formulário.
- Preencherá seus respectivos campos, podendo selecionar o estilo, o tipo de dado e o tipo de campo. Se optar por tipo de campo, que não seja texto, ou seja, tipo select/checkbox/radio, aparecerá um campo com o nome "opções", no qual o usuário terá que digitar as opções separadas por vírgulas referentes ao campo.
- Após escolher todos os dados referentes ao novo formulário em (DV6), o usuário deverá pressionar sobre o botão "Adicionar".
- O sistema salva o novo template e na interface (DV3) abrirá uma nova aba com o novo formulário criado.
- Se o usuário desejar visualizar o formulário, ao passar o mouse sobre este poderá clicar no ícone "Visualizar" que abrirá a interface (DV5) instanciando o respectivo formulário. Lá ele poderá novamente editá-lo pressionando o ícone "Editar".
- Caso o usuário desejar cancelar a ação de incluir novo formulário em (DV6) deverá pressionar sobre o botão "*Cancelar*". O sistema deve retornar à tela "*Formulários*".
- O usuário poderá adicionar mais campos pressionando o ícone "Adicionar campo" onde novamente a janela (DV6) "Adicionar campo:propriedades" abrirá, e o usuário preencherá novamente seus campos, ou excluí-los, clicando sobre o ícone "Excluir" na aba referente a determinado formulário.
- Ao pressionar em "Excluir" uma mensagem de aviso aparecerá para o usuário confirmar se deseja mesmo excluir o campo. Após o usuário clicar no botão "Ok", aparecerá uma mensagem confirmando a exclusão do campo.

- Na interface (DV3) “Formulários” na aba ao lado de cada respectivo *Formulário*, o usuário realiza as configurações necessárias ligadas ao formulário clicando nos ícones editar/preencher/gerar relatório.

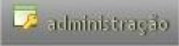
### **Cenários Secundários:**

- Algum campo inválido ou configurado incorretamente, sistema retorna mensagem de aviso. O usuário deve configurar corretamente todos os campos.
- Se o usuário deseja editar os campos de um formulário ele deve pressionar o ícone “ editar”. O sistema abre a tela (DV4) “*Cadastro de um Novo Formulário*” que conterà na tabela “Propriedades do Formulário e dos Campos” os campos referentes ao respectivo formulário assim como o ícone “ adicionar campo” e o ícone “ excluir”.
- Se o usuário deseja excluir um campo do formulário ele deve pressionar no ícone “ excluir” na linha do respectivo campo que será excluído. O sistema apresenta uma mensagem de confirmação de exclusão, o usuário confirma e o campo do formulário é deletado.
- Se o usuário deseja somente adicionar mais um campo ele deverá pressionar novamente o “ícone “ adicionar campo” onde abrirá a janela(DV6) “Adicionar campos:propriedades” onde novos campos podem ser adicionados e salvos gerando assim um novo template.



### **Fluxo de Eventos**

#### **Fluxo principal:**


#### **P1. Incluir Formulário**




P1.1 O caso de uso se inicia quando o ator acessar a interface principal do sistema (DV1) e clicar no ícone “ administração” abrindo a janela de Login (DV2) onde o usuário digitará login e senha.


P1.2 O sistema chama o <Include – Efetuar Login>.


P1.3 Após o usuário estar logado ao sistema, no menu superior o ícone “Status:deslogado”  mudará seu estado para “Status: logado”  ”.


P1.4 Ao usuário pressionar o item de menu "Formulário", O sistema apresentará a interface (DV3) “Formulários”

P1.5 A interface (DV3) “Formulários”, contém os respectivos formulários anteriormente cadastrados e o ícone “Novo”  .

P1.6 Ao usuário passar o mouse em cima de determina formulário aparecerão os ícones”  ”(Editar),”  ”(Visualizar) e “  ”(relatório).

P1.7 O ator poderá pressionar o ícone “Novo”  e o sistema redirecionará para a interface “ Cadastro de um novo formulário”(DV4);(A1)(A2)


P1.8 Na interface (DV4) o usuário preenche o nome e a descrição do formulário e clica no botão “Submeter”  .(E1)

P1.9 Caso o usuário pressione o ícone “  adicionar campo ”o sistema chama <UC – Manter Campo>.


## Fluxos Alternativos:

### A1.Editar Formulário


Em P1, se o usuário optar por Editar um formulário já existente no curso:

A1.1 O usuário escolhe o formulário já existente a ser alterado na interface(DV3)”Formulários” e pressiona o ícone “ ”.

A1.2 O sistema abre a interface (DV4) “*Cadastro de um Novo Formulário*” com os dados instanciados referente ao formulário selecionado podendo assim os dados serem editados.


A1.3 O usuário altera a informação dos campos “nome” e/ou “descrição” e pressiona o botão “Submeter”  .


A1.4 O sistema salva os dados e exibe as alterações na tela(DV4).

A1.5 Caso o usuário pressione o ícone “ ”, o sistema chama <UC – Manter Campo>.

## A2. Visualizar Formulário

Em P1, se o usuário optar por Visualizar um formulário já existente :

A2.1 O usuário escolhe o formulário a ser visualizado e clica no ícone “ ”(Visualizar).

A2.2 O sistema abre a interface (DV5) onde o formulário poderá ser visualizado e na aba com os dados do formulário o usuário também terá a opção “ ”(Editar) caso queira novamente editar o formulário.

## Fluxos de Exceção

### E1. Campos obrigatórios não preenchidos

- Ao usuário cadastrar um novo formulário na interface(DV4) caso um dos campos “nome” ou “descrição” não sejam preenchidos,o sistema emite um aviso :“O campo nome deve ser preenchido” e/ou “O campo descrição deve ser preenchido”.
- Na interface (DV6)” “Adicionar campos:propriedades” caso o usuário não digite o nome do rótulo do campo e da descrição,o sistema produzirá o seguinte aviso:”O campo rótulo deve ser preenchido” e/ou “O campo descrição deve ser preenchido”.
- Também na interface (DV6),caso o usuário escolha o tipo de campo(Select/Checkbox/Radio)o campo opções aparecerá na tela e

deverá ser preenchido. Se não for, o sistema emitirá o seguinte aviso: "O campo opções deve ser preenchido se o tipo de campo selecionado for select, radio ou checkbox".

- Na interface de Login(DV2) se o usuário esquecer de preencher campo de login ou senha, o sistema produzirá os seguintes avisos: "O campo login deve ser preenchido" e/ou "O campo senha deve ser preenchido".

**Pós-Condições:** Após o término deste caso de uso o administrador poderá editar, alterar e visualizar um formulário criado.

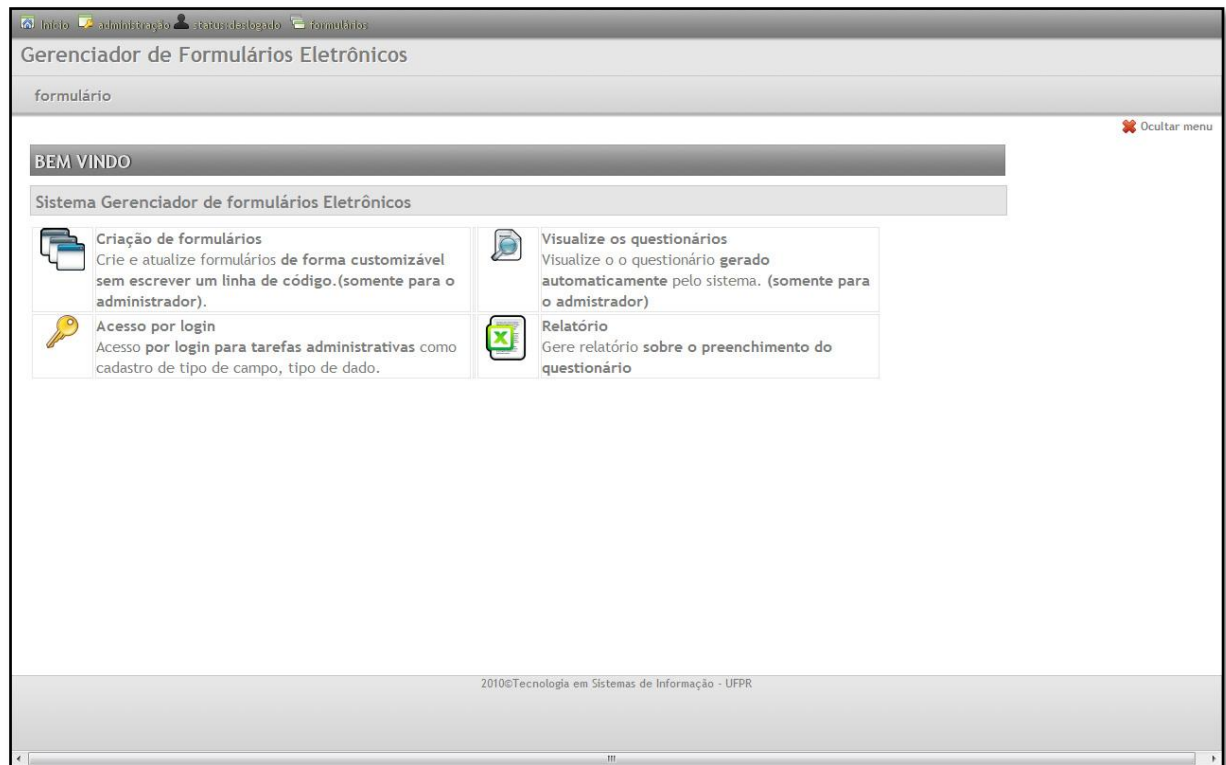
**Pontos de extensão:** Não há.

### **Requisitos Não Funcionais**

- Para edição de um formulário é permitida a seleção de um formulário de cada vez.
- Para a inclusão de um novo formulário só é permitida a criação de um formulário por vez.

#### **4.2.1.1. TELAS**

DV1: tela inicial



**Figura 7 DV1 - tela inicial**

DV2: tela de login

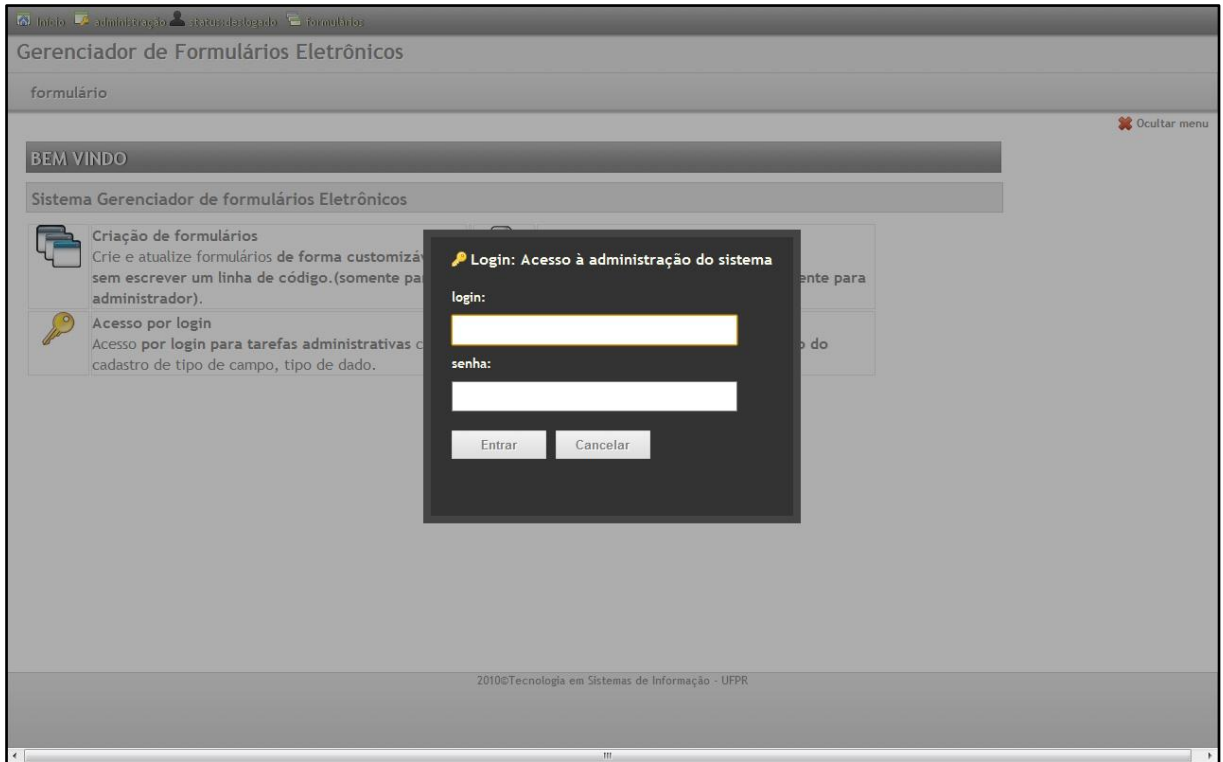


Figura 8 DV2 - tela de login

DV3:Tela Formulários (deslogado)

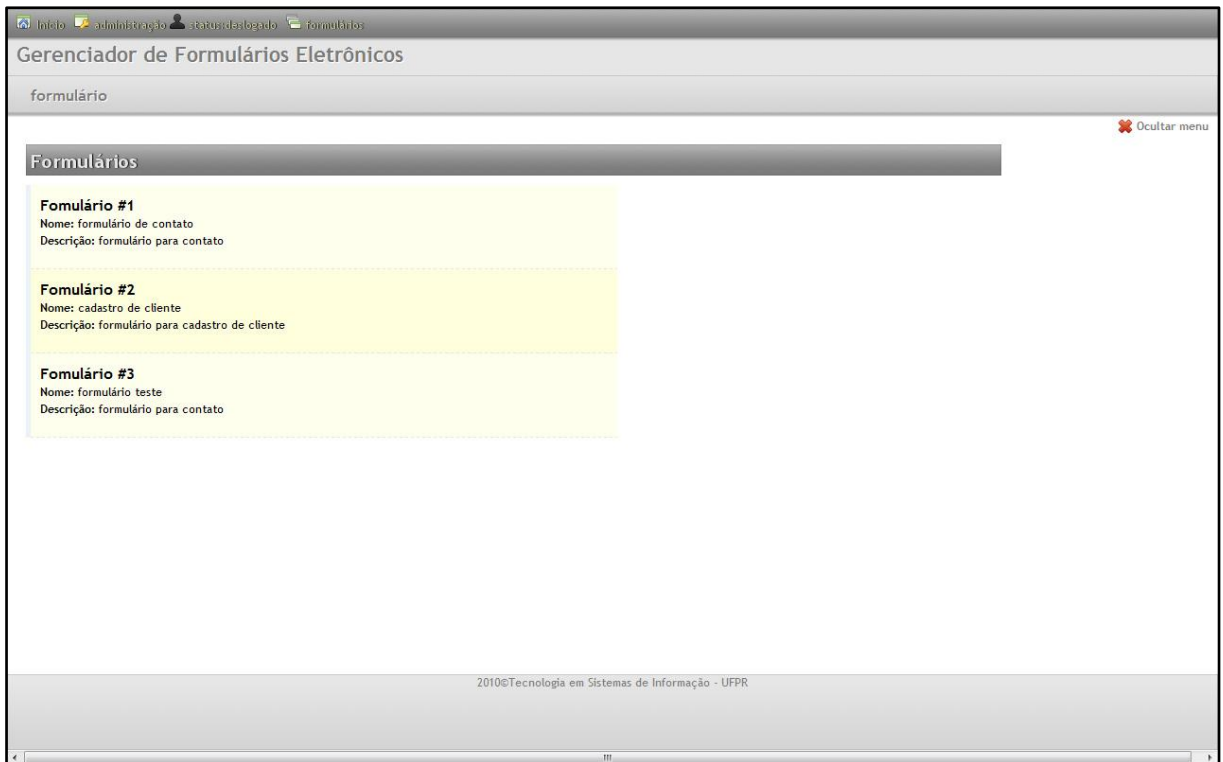


Figura 9 - DV3 - tela formulários (deslogado)



### DV3: Tela Formulários (logado)

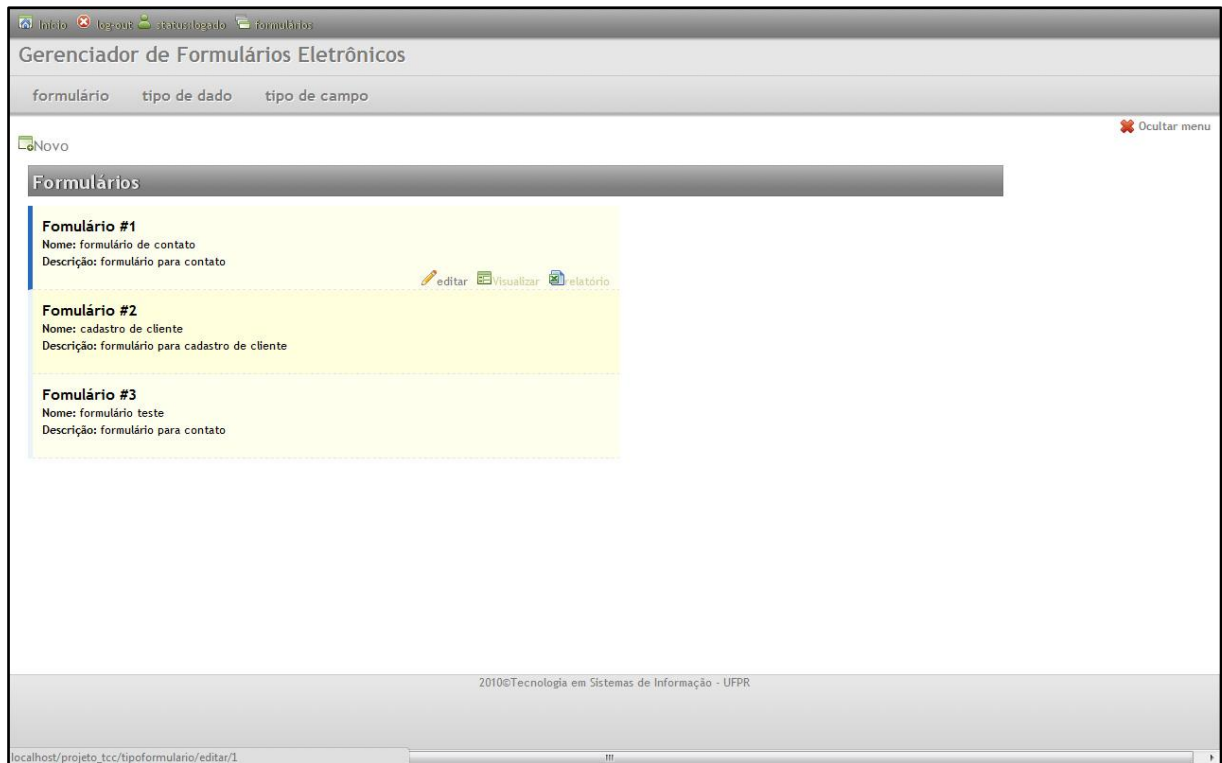


Figura 10 DV3 - tela formulários (logado)

### DV4: Tela Cadastro de um novo formulário

Gerenciador de Formulários Eletrônicos

formulário tipo de dado tipo de campo

Ocultar menu

Cadastro de um novo Formulário

nome:

descrição:

Submeter Cancelar

2010: Tecnologia em Sistemas de Informação - UFPR

Figura 11 - DV4 - Tela cadastro de um novo formulário

## DV4 – Tela Edição do Formulário

Gerenciador de Formulários Eletrônicos

formulário tipo de dado tipo de campo

Ocultar menu

Edição do formulário

nome:

cadastro de cliente

descrição:

formulário para cadastro de cliente

Submeter Cancelar

Propriedades do formulário e dos campos

**Formulário #2**  
Nome: cadastro de cliente  
Descrição: formulário para cadastro de cliente

Adicionar campo

Rótulo	Descrição	Estilo	Ordem	Obrigatorio	Ação
nome	campo para nome	grande	1	não	excluir editar
cidade	campo para cidade	pequeno	1	não	excluir editar
idade	campo para a idade	pequeno	1	não	excluir editar
e-mail	campo para e-mail	pequeno	1	sim	excluir editar
sobrenome	campo para o sobrenome	pequeno	2	não	excluir editar
estado	descricao	pequeno	3	sim	excluir editar

2010: Tecnologia em Sistemas de Informação - UFPR

Figura 12 DV4 - Tela Edição do formulário

## DV5: Tela visualiza formulário

**Fomulário 2**  
Nome: cadastro de cliente  
Descrição: formulário para cadastro de cliente

editar

**cadastro de cliente**

nome

cidade  idade  e-mail

sobrenome  estado

submeter

localhost/projeto\_tcc/tipoformulario/editar/2

**Figura 13 DV5 tela visualiza formulário**

### 4.2.1.2. DIAGRAMA DE SEQUENCIA: INCLUIR FORMULÁRIO

Conforme demonstrado no apêndice B.

### 4.2.1.3. DIAGRAMA DE SEQUENCIA: ALTERAR FORMULÁRIO

Conforme demonstrado no apêndice C.

### 4.2.2. CASO DE USO: MANTER CAMPO

**Ator:** Administrador

**Visão geral:** Este caso de uso tem por objetivo permitir ao administrador do sistema a inclusão, alteração e exclusão de Campo(s) do Formulário.

**Estado:** Especificado

**Pré-Condições:**

O usuário deve ter selecionado um formulário.

**Cenário Principal**

- Após o usuário se logar no sistema e pressionar no menu “Formulário”, o sistema carrega a interface(DV3) “Formulários”.
- O usuário em (DV3) pressiona o ícone “Novo” e o sistema abre a interface “Cadastro de um Novo/Edita Formulário”(DV4).
- Em (DV4) o usuário preenche o nome e a descrição do formulário e clica no botão “Submeter”.
- Abre um painel em (DV4) denominado “Propriedades do campo do formulário” contendo os dados do formulário salvo e o ícone “Adicionar Campos”.
- Ao o usuário pressionar o ícone “Adicionar campo” o sistema redirecionará para a interface(DV6) “Adicionar campo:propriedades” onde novos campos podem ser adicionados e salvos gerando assim um novo template.
- Caso o usuário queira excluir algum campo irá na interface(DV4) no painel “Propriedades do campo do Formulário”, onde para cada campo temos uma linha da tabela com um ícone “Excluir”.
- Ao pressionar o ícone “Excluir” uma caixa de confirmação será exibida, e se a opção “Ok” for escolhida o registro será imediatamente deletado e excluído da tabela, assim como aparecerá uma mensagem de feedback confirmando a operação.


**Cenários Secundários:**

- O usuário deve configurar corretamente os dados do campo.

**Fluxo Principal**

## P1. Incluir Campo


P1.1 O usuário abre a tela “ Cadastro de Novo/Edita Formulário” na interface (DV4).

P1.2 O usuário pressiona o ícone “Adicionar campo”  .

P1.3 O sistema exibe a janela “Adicionar campos:Propriedades”(DV6).

P1.4 O usuário preenche o Rótulo a descrição e faz as configurações relacionadas ao campo.

P1.5 Se o tipo de campo escolhido for Select/Radio ou Checkbox,abrirá na tela(DV6) um campo opção, onde o usuário deverá preencher os valores que serão utilizados na construção do campo,sendo que estes devem ser digitados separados por vírgulas.


P1.6 O usuário entra com os dados e depois pressiona o botão “  ”e os novos campos serão salvos.(E1)

P1.7 O sistema inclui registro no grid.(A1)(A2)

## Fluxos Alternativos

### A1. Alterar Campo

Em P2, o usuário pode optar em alterar o campo incluído:

A1.1 O usuário seleciona o campo que deseja alterar no Grid da tabela na interface(DV7)”Cadastro de um novo tipo de campo” e pressiona o ícone “  ” do respectivo campo que será alterado.

A1.2 O usuário faz as configurações necessárias.

A1.4 O usuário pressiona o botão “Submeter”  da aba.

A1.5 O sistema salva as configurações do campo.

A1.6 O sistema volta para o <Extend Manter Formulário >

### A2. Excluir Campo

Em P2, o usuário pode optar por excluir o campo incluído:

A2.1 O usuário seleciona o campo, na interface(DV4)”Cadastro de um novo Formulário”.

A2.2 O dado que será alterado, está visível na mesma linha no Grid da tabela na aba “Nome”, ao lado do botão “Excluir”.

A2.3 O usuário pressiona o ícone “excluir”.

A2.4 O sistema emite uma mensagem de confirmação de exclusão: “Ao excluir esse campo todas as respostas vinculadas a ele serão deletadas. Deseja mesmo excluí-lo?”. Se o usuário quiser mesmo fazer a exclusão, ele deve clicar no botão “Ok”, caso queira cancelar clica no botão “Cancelar”.

A2.5 O sistema exclui o campo e emite uma mensagem de feedback “Campo excluído com sucesso”.

A2.6 O sistema volta para o <Extend Manter Formulário >

## Fluxos de Exceção

### E1. Campos obrigatórios não preenchidos

- Caso o usuário escolha na interface(DV6) o tipo de campo como Select/Radio/Checkbox ,e não preencher o campo opções relativo aos valores do campo, o sistema emitirá um aviso: “O campo opções deve ser preenchido se o tipo de campo selecionado for select, radio ou checkbox”.

## Pós-condições

Ao concluir este caso de uso o administrador poderá incluir, alterar ou excluir um campo.

## Requisitos Não Funcionais

- Para inclusão, alteração ou exclusão é permitida a seleção de um campo de cada vez.
- Facilidade de navegação.

### 4.2.2.1. TELAS

## DV4: Cadastro de Novo/Edita formulário

**Gerenciador de Formulários Eletrônicos**

formulário   tipo de dado   tipo de campo

Ocultar menu

**Edição do formulário**

nome:  
cadastro de cliente

descrição:  
formulário para cadastro de cliente

Submeter   Cancelar

**Propriedades do formulário e dos campos**

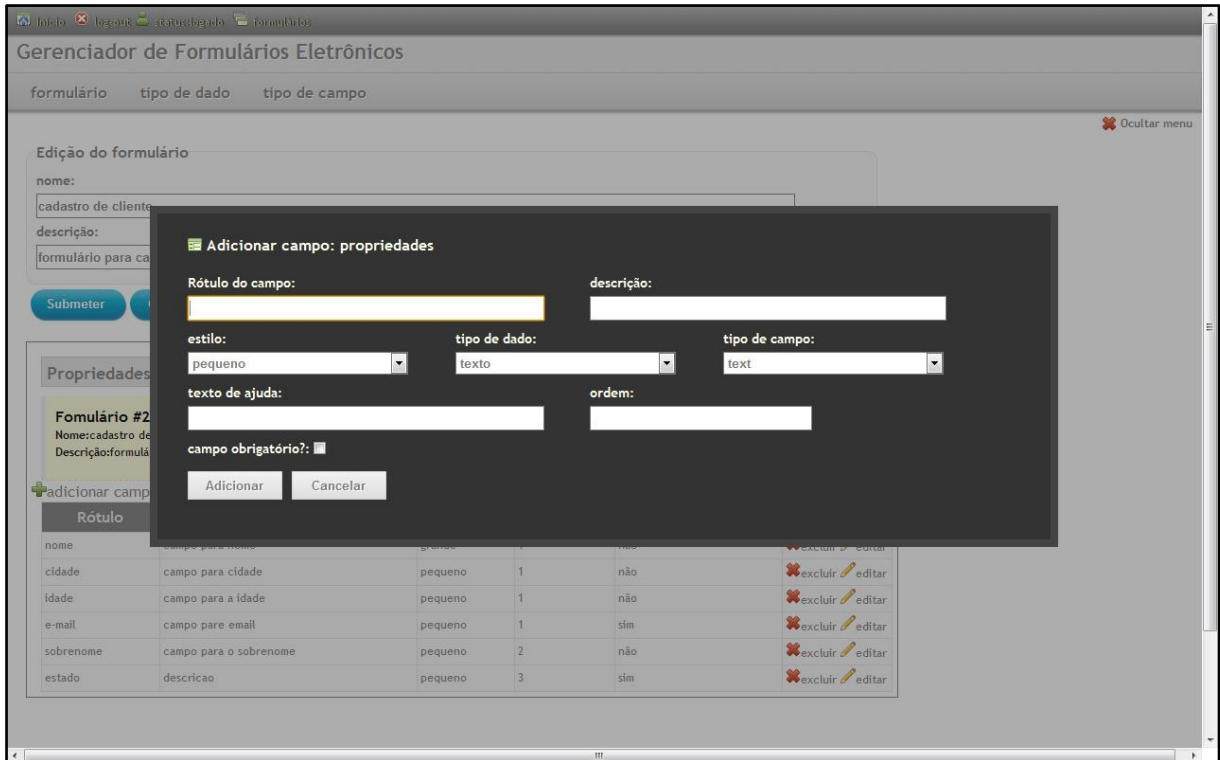
**Formulário #2**  
Nome:cadastro de cliente  
Descrição:formulário para cadastro de cliente

+ Adicionar campo

Rótulo	Descrição	Estilo	Ordem	Obrigatorio	Ação
nome	campo para nome	grande	1	não	excluir / editar
cidade	campo para cidade	pequeno	1	não	excluir / editar
idade	campo para a idade	pequeno	1	não	excluir / editar
e-mail	campo para email	pequeno	1	sim	excluir / editar
sobrenome	campo para o sobrenome	pequeno	2	não	excluir / editar
estado	descricao	pequeno	3	sim	excluir / editar

Figura 14 DV4 cadastro de novo/edita formulário

## DV6: Adicionar Campo: Propriedades



**Figura 15 DV6 tela adiciona campo:propriedades**

#### 4.2.2.2. DIAGRAMA DE SEQUENCIA: INCLUIR CAMPO

Conforme demonstrado no apêndice D.

#### 4.2.2.3. DIAGRAMA DE SEQUENCIA : ALTERAR CAMPO

Conforme demonstrado no apêndice E.

#### 4.2.2.4. DIAGRAMA DE SEQUENCIA: EXCLUIR CAMPO

Conforme demonstrado no apêndice F.

### 4.2.3. CASO DE USO: GERAR RELATÓRIO

**Atores:** Administrador.

**Visão geral:** Este caso de uso tem por objetivo permitir ao administrador visualizar os questionários preenchidos pelos usuários do sistema .




**Estado:** Especificado

**Pré-Condições:**

O usuário deve ter selecionado um formulário.

**Cenário Principal:**


- Se o usuário deseja visualizar um formulário ele deve pressionar o ícone “relatório” do formulário. O sistema retorna uma tela com a visualização do layout do formulário.

**Cenários Secundários:**

- Para voltar para a interface(DV3)”Formulários” o usuário deve fechar o arquivo.

## Fluxo Principal

### P1. Gerar Formulário

- P1.1 Na interface (DV3) “Formulários”, o usuário do sistema pressiona o ícone “Relatório” “relatório”.
- P1.2 O sistema abrirá o arquivo referente aos formulários preenchidos.
- P1.3 O usuário após visualizar as informações fecha a página.
- P1.4 O sistema volta ao <Extend Manter Formulário>.

## Fluxos Alternativos

Nenhum fluxo alternativo encontrado.

## Fluxos de Exceção

Nenhum fluxo de exceção encontrado.

## Pós-condições

Foi visualizado os questionários.

## Requisitos Não Funcionais

- Para visualização é permitida a seleção de um determinado formulário de cada vez.

### 4.2.3.1. DIAGRAMA DE SEQUENCIA: GERAR RELATÓRIO

Conforme demonstrado no apêndice G.

### 4.2.4. CASO DE USO: MANTER TIPO DE CAMPO

**Ator:** Administrador.

**Visão geral:** Este caso de uso tem por objetivo permitir ao administrador configurar os tipos de campos que serão utilizados no sistema podendo Incluir, Alterar e Excluir Campo(s).

**Estado:** Especificado

#### **Pré-Condições:**

O usuário deve ter a necessidade de configurar um tipo de campo.

O usuário ser o administrador e estar logado no sistema.

#### **Cenário Principal:**

- O usuário abre a tela *“inicial” do sistema(DV1)* e pressiona o ícone *“Administração”* e efetua o Login na interface(DV2).

- O sistema chama o <Include – Efetuar Login>.
- O usuário após logado no sistema pressiona na tela (DV1) em “formulários”.
- O sistema redireciona para a interface(DV3)”Formulários”.
- Em (DV3) o usuário seleciona no menu principal “Tipo de campo”.
- O sistema abrirá a interface(DV7) ”Cadastro de um novo tipo de campo”.
- O usuário poderá incluir um novo tipo de campo, preenchendo o campo “nome” e ao pressionar o botão “Submeter” o novo campo criado aparecerá na tabela na interface(DV7).
- Nesta tabela cada linha tem um “Id” e um “Nome” relacionado a cada respectivo tipo de campo, assim como os ícones ”Excluir” e “Editar”.
- Se o usuário quiser excluir um registro, deverá pressionar o ícone “Excluir” onde abrirá a janela de mensagem: ”Se este tipo de campo estiver sendo referenciado ele não será excluído.Deseja mesmo excluí-lo?”. Caso o usuário confirme a exclusão, o sistema retornará uma mensagem de feedback: ”Campo Excluído com Sucesso”.
- Caso o usuário desejar alterar um tipo de campo, ele poderá pressionar na linha do determinado registro o ícone “Editar”, e em “Cadastrar um novo tipo de campo”, o usuário preencherá no campo nome, o novo registro a ser inserido.
- Ao pressionar o botão “Submeter” o tipo de campo será modificado na tabela na interface(DV7). Se o usuário desistir de alterar o campo ele poderá pressionar o botão “Cancelar”.

#### **Cenários Secundários:**

- O usuário deve configurar corretamente os dados do campo.


### **Fluxo Principal**

#### **P1. Incluir Tipo de Campo**

P1.1 O usuário pressiona, no menu principal, o link “Tipo de campo”.

P1.2 O sistema abre a interface (DV7).

P1.3 O usuário na interface(DV7) "Cadastro de um novo tipo de campo" preenche os campos nome.

P1.4 E pressiona o botão "Submeter"  . (E1)

P1.5 O sistema salva os dados e emite um aviso:"Tipo de campo salvo com sucesso".


P1.6 O novo registro será incluído no grid da interface(DV7).(A1)(A2)

P1.7 O usuário poderá novamente repetir a operação caso queira adicionar mais algum tipo de campo.

## Fluxos Alternativos

### A1. Alterar Tipo de Campo

Em P1, o usuário pode optar por alterar o dado incluído:

A1.1 O usuário seleciona o registro que deseja alterar no grid da tabela na interface(DV7) e clica no ícone "  " do respectivo tipo de campo que será alterado.

A1.2 O sistema preenche os campos do formulário.

A1.3 O usuário altera os valores dos campos no formulário na interface(DV7) "Cadastro de um novo tipo de campo".

A1.4 O usuário pressiona o botão "Submeter".

A1.5 O sistema salva as configurações do tipo de campo e inclui o registro no grid da tabela.

A1.6 O sistema emite uma mensagem de retorno: "Tipo de campo salvo com sucesso".

### A2. Excluir Tipo de Campo

Em P1, o usuário pode optar por excluir o tipo de campo incluído:

A2.1 O usuário seleciona o registro no Grid da tabela na interface(DV7)".

A2.2 E na linha do tipo de campo que deseja excluir, pressiona o ícone "



A2.3 O sistema abre uma janela de mensagem : "Se este tipo de campo estiver sendo referenciado ele não será excluído.Deseja mesmo excluí-lo?".

A2.4 O usuário pressiona o botão “Ok” confirmando a exclusão.

A2.5 O sistema exclui o tipo de campo e exclui o registro no grid.

A2.6 O sistema emite uma mensagem de feedback:” Tipo de campo excluído com sucesso” e a operação é concluída.

## **Fluxos de Exceção**

### **E1. Campos obrigatórios não preenchidos**

- Caso o usuário não preencha Na interface(DV7) em”Cadastro de um novo tipo de campo” o nome do campo a ser incluído ou alterado,o sistema produzirá um aviso:”O campo nome deve ser preenchido”.

### **Pós-condições**

Ao concluir este caso de uso o administrador poderá incluir, alterar ou excluir um tipo de campo.

## **Requisitos Não Funcionais**

- Para inclusão, alteração ou exclusão é permitida a seleção de um campo de cada vez.

### **4.2.4.1. TELAS**

DV1: Tela inicial

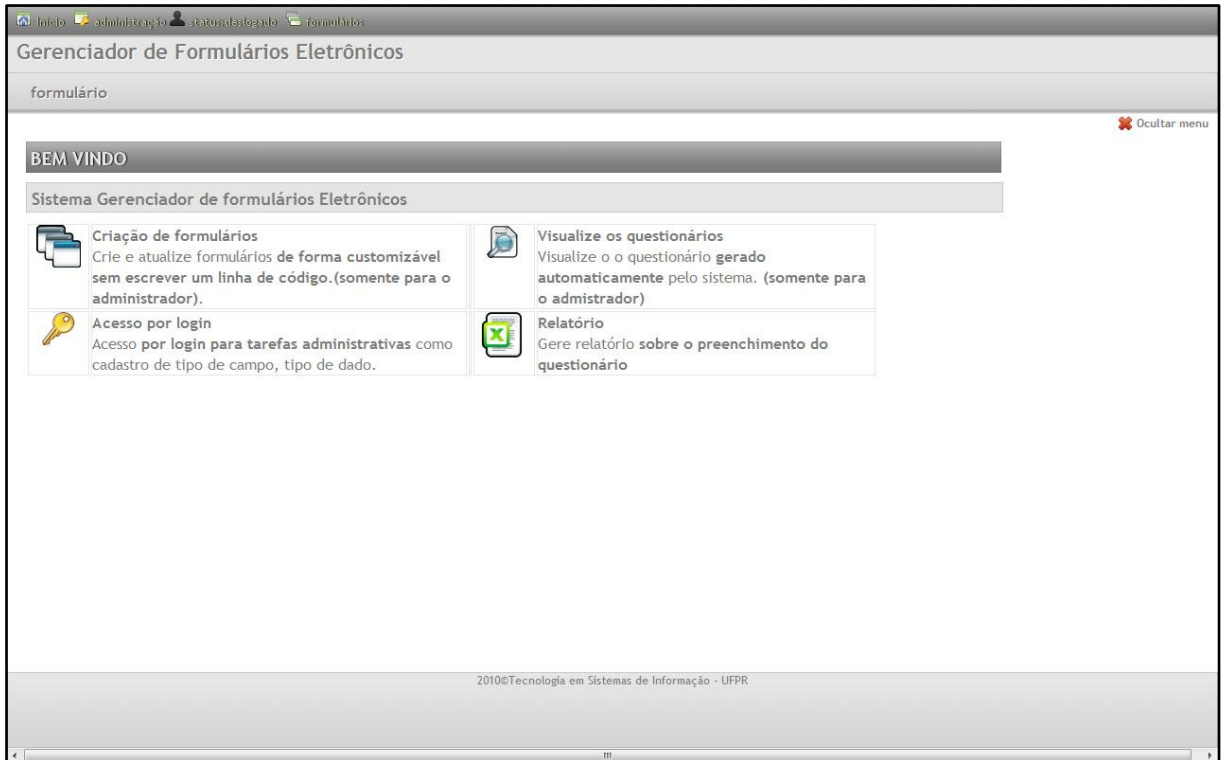


Figura 16 DV1 Tela inicial

## DV2: Login

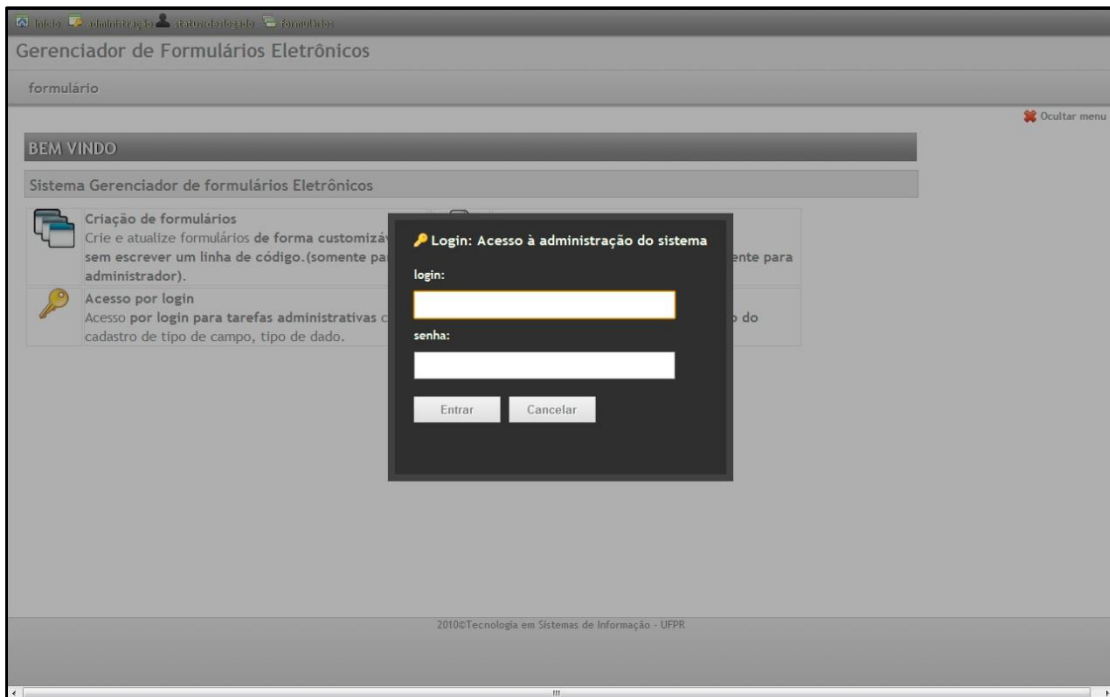


Figura 17 DV2 tela de login

## DV7: Cadastro de tipo de campo

Gerenciador de Formulários Eletrônicos

formulário tipo de dado tipo de campo

cadastro de um novo tipo de campo

nome:

Submeter Cancelar

id	Nome	Ação
1	text	excluir editar
2	password	excluir editar
3	radio	excluir editar
4	checkbox	excluir editar
5	select	excluir editar
6	textarea	excluir editar

2010©Tecnologia em Sistemas de Informação - UFPR

### 4.2.4.2. DIAGRAMA DE SEQUENCIA: INCLUIR TIPO DE CAMPO

Conforme demonstrado no apêndice H.

### 4.2.4.3. DIAGRAMA DE SEQUENCIA: ALTERAR TIPO DE CAMPO

Conforme demonstrado no apêndice I.

### 4.2.4.4. DIAGRAMA DE SEQUENCIA: EXCLUIR TIPO DE CAMPO

Conforme demonstrado no apêndice J.

### 4.2.5. CASO DE USO: MANTER TIPO DE DADO

**Ator:** Administrador.

**Visão geral:** Este caso de uso tem por objetivo permitir ao administrador configurar os tipos de dados que serão utilizados no sistema podendo Incluir, Alterar e Excluir Dado(s).

**Estado:** Especificado

**Pré-Condições:**

O usuário deve ter a necessidade de configurar um tipo de dado.

O usuário deve ser o administrador e estar logado no sistema.

**Cenário Principal:**

- O usuário abre a tela *“inicial” do sistema(DV1)* e clicando no ícone “Administração” efetua o Login na interface(DV2).
- O sistema chama o <Include – Efetuar Login>.
- O usuário após logado no sistema pressiona na tela (DV1) em “Formulários”.
- O sistema redireciona para a interface(DV3)”Formulários”.
- Em (DV3) o usuário seleciona no menu principal “Tipo de dado”.
- O sistema abrirá a interface(DV8)”Cadastro de um novo tipo de dado”.
- O usuário poderá incluir um novo tipo de dado,preenchendo os campos nome e descrição e pressionar o botão “Submeter”.
- O sistema emitirá um aviso:”Tipo de dado salvo com sucesso”.
- O novo dado será salvo e ficará visível no grid da tabela na interface(DV8).
- Nesta tabela cada linha tem um “Id”, “Nome” e “Descrição”, relacionado a cada respectivo dado,assim como os ícones”Excluir” e “Editar”.
- Se o usuário optar por excluir um dado,deverá pressionar o ícone “Excluir” onde abrirá a janela de mensagem: ”Se este tipo de dado estiver sendo referenciado ele não será excluído.Deseja mesmo excluí-lo?”.
- Se o usuário confirmar a exclusão,o sistema retorna uma mensagem de feedback:”Tipo de dado excluído com sucesso”.
- No caso de o usuário desejar alterar um dado, ele poderá selecionar a linha do dado escolhido e pressionar no ícone “Editar” e inserir no painel



“Cadastrar um novo tipo de dado”, o novo nome e descrição do dado a ser alterado.

- O usuário preenche os novos dados e pressiona o botão “Submeter”. Caso o usuário desista de alterar o dado, ele poderá pressionar o botão “Cancelar”.
- Ao usuário clicar em “Submeter”, o sistema emitirá um aviso: “Tipo de dado salvo com sucesso”.
- O novo dado será modificado e ficará visível no grid da tabela na interface(DV8).

### **Cenários Secundários:**

- O usuário deve configurar corretamente o tipo dos dados.


## **Fluxo Principal**

### **P1. Incluir Tipo de Dado**

P1.1 O usuário pressiona, no menu principal, o link “Tipo de dado”.

P1.2 O sistema abre a interface (DV8).

P1.3 O usuário na interface(DV8) “Cadastro de um novo tipo de dado” preenche os campos nome e descrição.

P1.4 E pressiona o botão “Submeter” . (E1)

P1.5 O sistema salva os dados e emite um aviso: “Tipo de dado salvo com sucesso”.


P1.6 O novo registro será incluído no grid da interface(DV8).(A1)(A2)

P1.7 O usuário poderá novamente repetir a operação caso queira adicionar mais algum tipo de dado.

## **Fluxos Alternativos**

### **A1. Alterar Tipo de Dado**

Em P1, o usuário pode optar por alterar o dado incluído:

A1.1 O usuário seleciona o registro que deseja alterar no Grid da tabela na interface(DV8) e clica no ícone “” do respectivo tipo de dado que será alterado.

A1.2 O sistema preenche os campos do formulário.

A1.3 O usuário altera os valores dos campos no formulário na interface(DV8) “Cadastro de um novo tipo de dado”.

A1.4 O usuário pressiona o botão “Submeter”.

A1.5 O sistema salva as configurações do tipo de dado e inclui o registro no grid da tabela.

A1.6 O sistema emite uma mensagem de retorno: “Tipo de dado salvo com sucesso”.

### **A2. Excluir Tipo de Dado**

Em P1, o usuário pode optar por excluir o tipo de dado incluído:

A2.1 O usuário seleciona o registro no Grid da tabela na interface(DV8)”.

A2.2 E na linha do tipo de dado que deseja excluir,pressiona o ícone “



”.

A2.3 O sistema abre uma janela de mensagem : “Se este tipo de dado estiver sendo referenciado ele não será excluído.Deseja mesmo excluí-lo?”.

A2.4 O usuário pressiona o botão “Ok” confirmando a exclusão.

A2.5 O sistema exclui o tipo de dado e exclui o registro no grid.

A2.6 O sistema emite uma mensagem de feedback:” Tipo de dado excluído com sucesso” e a operação é concluída.

## **Fluxos de Exceção**

### **E1. Campos obrigatórios não preenchidos**

- Se o usuário não preencher o campo nome e descrição do tipo do dado na interface (DV8) ao incluir ou alterar um tipo de dado,o sistema

emitirá os seguintes avisos: "O campo nome deve ser preenchido" e/ou "O campo descrição deve ser preenchido".

## Pós-condições

Ao concluir este caso de uso o administrador poderá incluir, alterar ou excluir um tipo de dado.

## Requisitos Não Funcionais

- Para inclusão, alteração ou exclusão é permitida a seleção de um tipo de dado a cada vez.

### 4.2.5.1. TELAS

#### DV1:Tela inicial

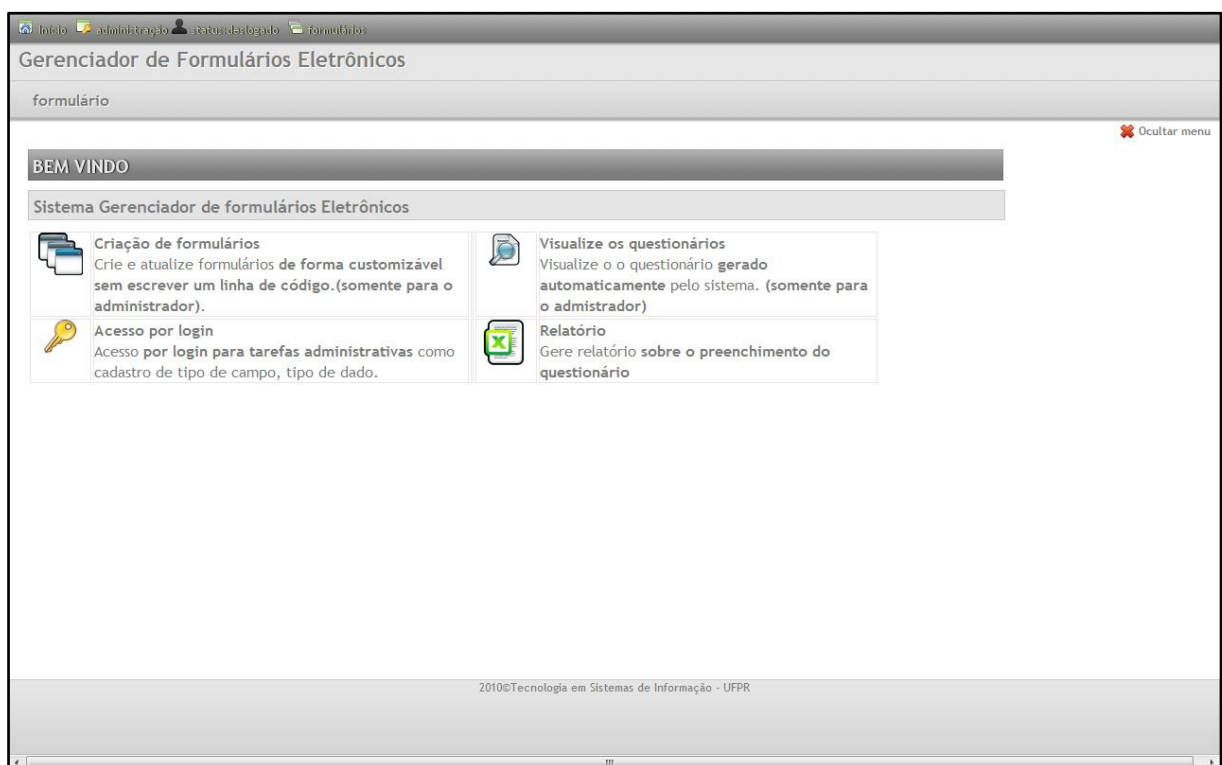


Figura 18 DV1 tela inicial

#### DV2: tela de login

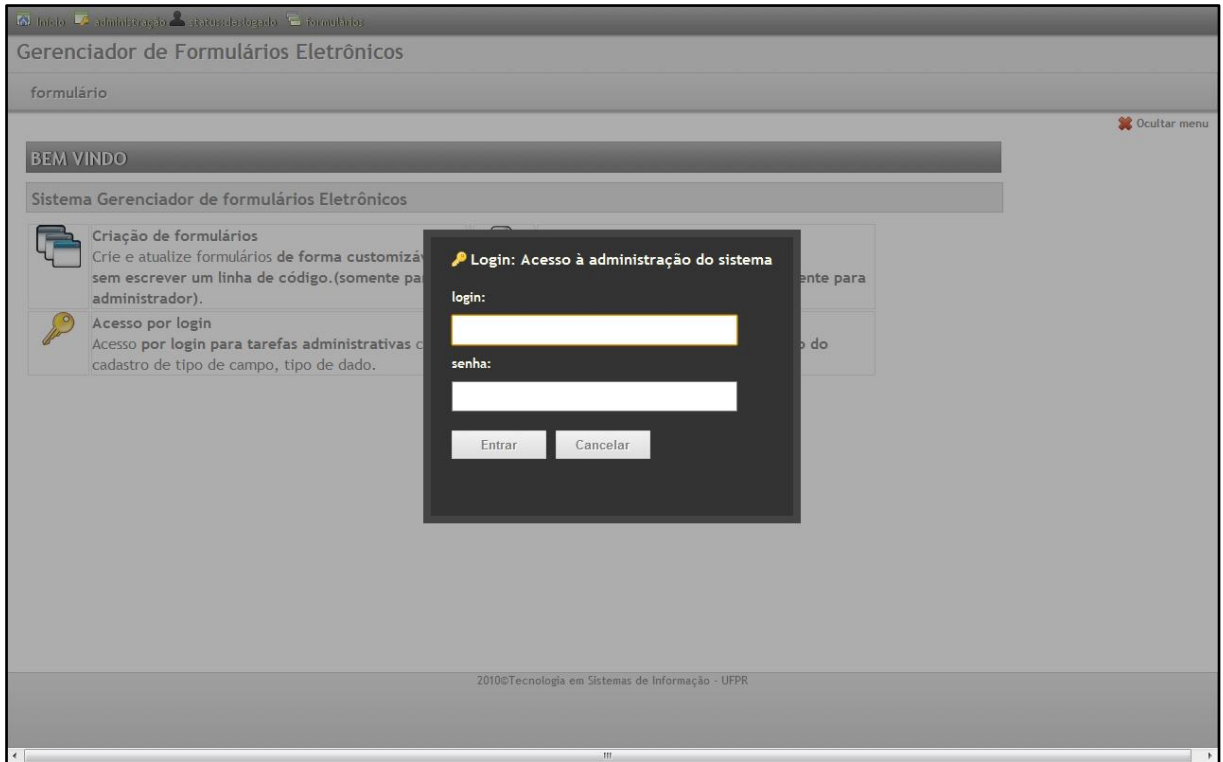


Figura 19 DV2 tela de login

DV8: cadastro de tipo de dado

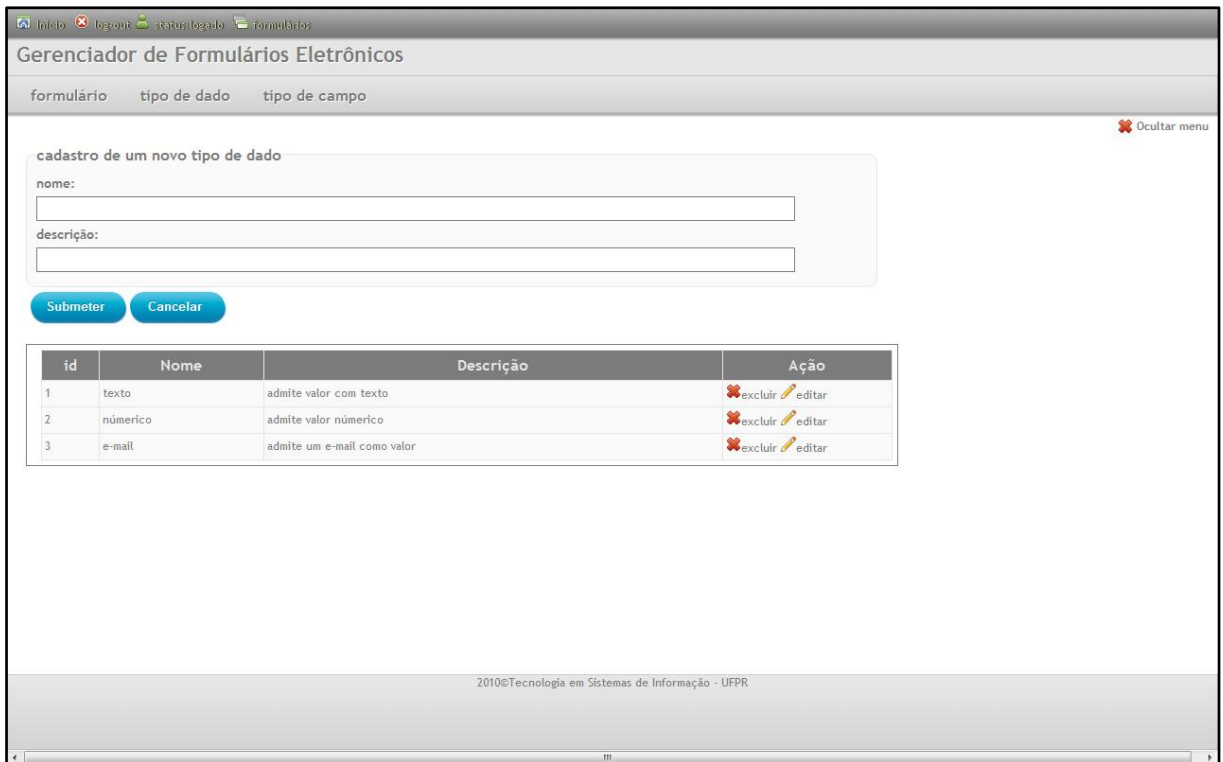


Figura 20 DV8 tela cadastro de tipo de dado

#### 4.2.5.2. DIAGRAMA DE SEQUENCIA: INCLUIR TIPO DE DADO

Conforme demonstrado no apêndice K.

#### 4.2.5.3. DIAGRAMA DE SEQUENCIA: ALTERAR TIPO DE DADO

Conforme demonstrado no apêndice L.

#### 4.2.5.4. DIAGRAMA DE SEQUENCIA: EXCLUIR TIPO DE DADO

Conforme demonstrado no apêndice M.

#### 4.2.6. CASO DE USO: PREENCHER FORMULÁRIO

**Ator:** Usuário.


**Visão geral:** Este caso de uso tem por objetivo permitir ao usuário do sistema escolher um formulário e preenche-lo.

**Estado:** Concreto

**Pré-Condições:**

O Formulário deve estar disponível para ser preenchido.

#### **Cenário Principal:**

- O usuário ao acessar a página inicial do sistema pressiona o link “Formulário” no menu.
- O sistema redireciona para a interface(DV3) que contém todos os formulários que foram cadastrados pelo administrador do sistema.
- Quando o usuário posicionar o mouse em cima de determinado formulário o ícone “Preencher” preencher aparecerá.

- Quando o usuário pressionar o ícone “preencher”, o sistema abrirá uma tela com o devido formulário.
- O usuário preenche os dados do formulário e pressiona no botão “Submeter”.
- O sistema salva os dados do usuário.

### **Cenários Secundários:**

- O usuário deve preencher todos os campos obrigatórios, caso contrário o sistema apresenta mensagem de erro.

### **Fluxo Principal**

P1. Preencher Formulário

P1.1 Na página inicial do sistema o usuário pressiona o link “Formulários”.

P1.2 O sistema abre a tela (DV3).

P1.3 O usuário escolhe um formulário e pressiona no ícone “preencher”.

P1.4 O sistema gera o questionário referente ao formulário e o exibe na tela.

P1.5 O usuário preenche os dados e pressiona o botão “Submeter” (E1)(E2).

P1.6 O sistema salva os dados, exibe mensagem de retorno e exibe a interface(DV3)”Formulários”.

### **Fluxos Alternativos**

Nenhum fluxo alternativo encontrado.

## **Fluxos de Exceção**

### **E1. Campos Obrigatórios**

E1.1 O sistema avisa que o(s) campo(s) obrigatório(s) não foi (foram) preenchido(s).

E1.2 O sistema seleciona o(s) campo(s) em questão.

E1.3 O usuário preenche o(s) campo(s) indicado(s).

### **E2. Campos Inválidos**

E2.1 O sistema avisa que o(s) campo(s) foi (foram) preenchido(s) de forma incorreta .

E2.2 O sistema seleciona o(s) campo(s) em questão.

E2.3 O usuário corrige o(s) campo(s) indicado(s).

## **Pós-condições**

Foi realizado o preenchimento de um formulário.

## **Requisitos Não Funcionais**

- Para o preenchimento de um formulário, é permitido preencher um formulário por vez.

### 4.2.6.1. TELAS

DV3: Formulários

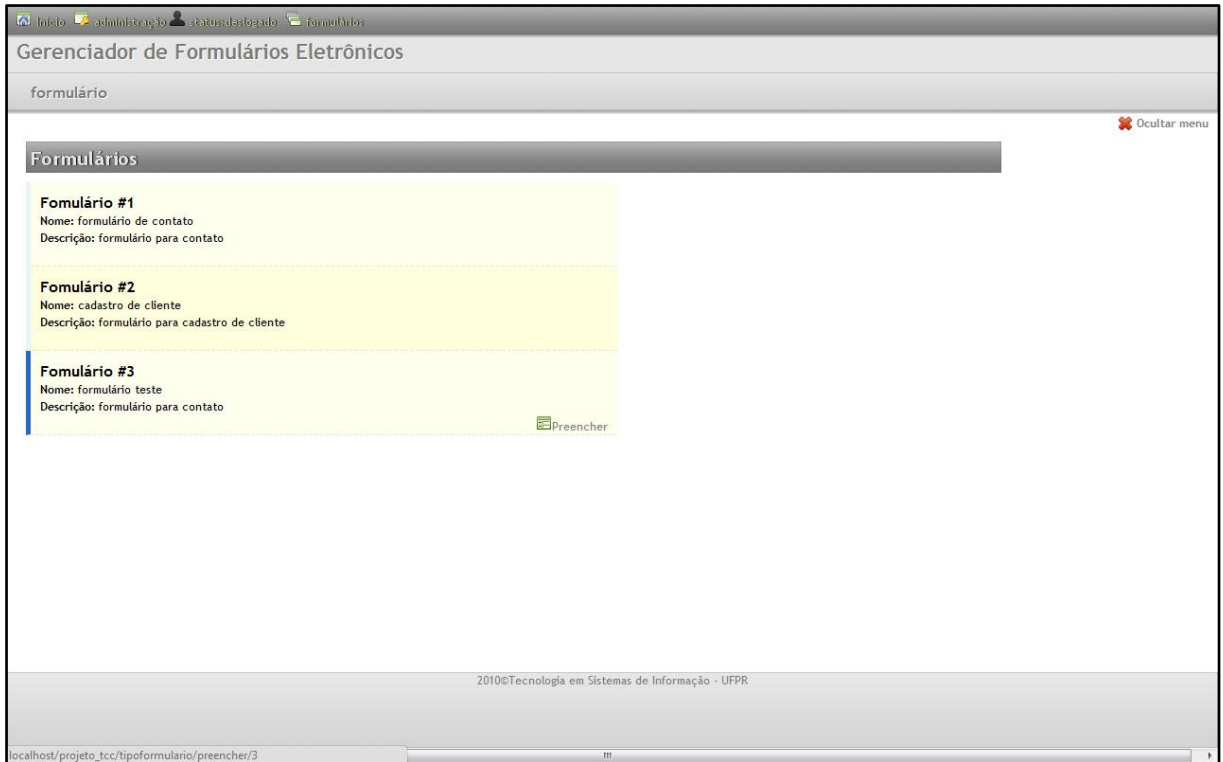


Figura 21 DV3: tela formulários (preencher)

DV9: tela do questionário

Figura 22 DV9 tela do questionário (gerado dinamicamente)



#### 4.2.6.2. DIAGRAMA DE SEQUENCIA: PREENCHER FORMULÁRIO

Conforme demonstrado no apêndice N.

#### 4.2.7. CASO DE USO: EFETUAR LOGIN

**Ator:** Administrador.





**Visão geral:** Este caso de uso tem por objetivo permitir ao administrador do sistema efetivar seu login.


**Estado:** Concreto

**Pré-Condições:**

O usuário possuir login e senha de administrador do sistema.

#### Cenário Principal:

- O usuário ao acessar a página inicial do sistema pressiona o ícone “” (Administração).
- O sistema abre a janela de Login (DV2) “Acesso à administração do sistema”.
- O usuário preenche os campos Login e senha.
- Após preencher os campos pressiona o botão “Entrar” para se logar ao sistema. E caso não queira mais se logar, poderá pressionar o botão “Cancelar”.
- O sistema efetivará o Login do usuário no sistema e emitirá a mensagem : “ Logado com sucesso.”.
- Automaticamente o ícone “” (Status:Deslogado) na barra de menu mudará seu estado para o ícone “”

(Status:Logado)”,assim como a seu lado o usuário encontrará o ícone “”(Logout)” que o usará quando quiser sair do sistema.

- O usuário estando logado poderá ter acesso as configurações do sistema.

#### **Cenários Secundários:**

- O usuário deverá preencher os dados de login e senha corretos senão o sistema apresenta mensagem de erro.

### **Fluxo Principal**

#### **P1. Preencher Login**

P1.1 Na página inicial do sistema(DV1) o usuário pressiona o ícone “



P1.2 O sistema abre a tela (DV2).(A1)

P1.3 O usuário digita o login e a senha e pressiona o botão”Entrar”.(E1)(E2)

P1.4 O sistema verifica se as informações de login estão corretas e emite aviso :”Usuário logado com sucesso”.

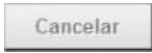
P1.5 O sistema retorna a página inicial do sistema(DV1) onde aparecerá uma mensagem de bem vindo ao Administrador.

P1.6 O usuário estando logado terá acesso às configurações do sistema.(A2)

### **Fluxos Alternativos**

#### **A1. Cancelar Login**

Em P1, se o usuário resolver cancelar sua entrada no sistema:

A1.1 O usuário pressiona o botão “Cancelar” .

A1.2 O sistema redireciona para a página inicial do sistema.

#### **A2. Efetuar Logout**

Em P1.6, se o usuário resolver sair do sistema:

A2.1 O usuário pressiona o ícone “Logout”.

A2.2 O sistema redireciona para a página inicial do sistema.

## Fluxos de Exceção

### **E1. Campos Obrigatórios**

E1.1 Se o usuário esquecer de digitar o campo de login ou senha.

E1.2 O sistema avisa que o(s) campo(s) obrigatório(s) não foram preenchido(s).

E1.2 O sistema seleciona o(s) campo(s) em questão.

E1.3 O usuário preenche o(s) campo(s) indicado(s).

### **E2. Acesso negado no Sistema**

E2.1 Se o usuário digitar o campo de login ou senha incorretos.

E2.2 O sistema emite um aviso :”Sem permissão de acesso”.

## **Pós-condições**

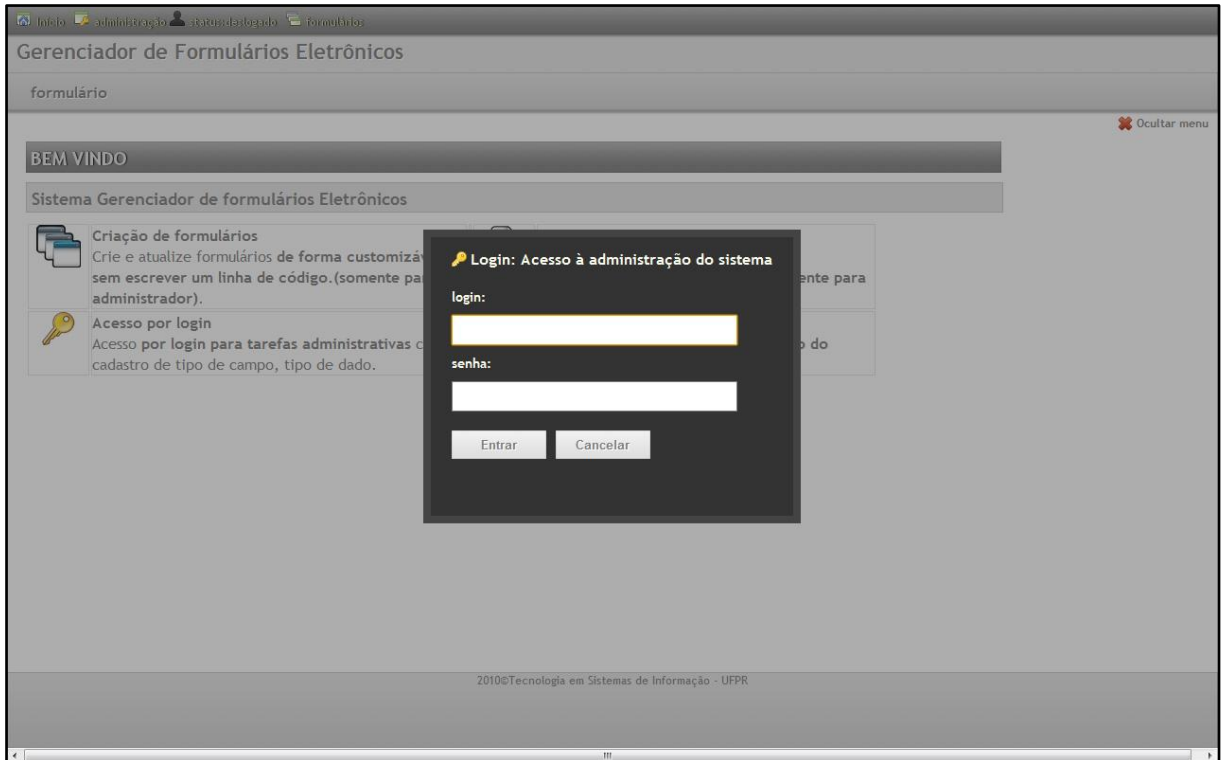
Foi realizado o Login do usuário no sistema.

## **Requisitos Não Funcionais**

- Para efetuar o Login o usuário deve possuir senha e login já cadastrados.

### 4.2.7.1. TELAS

DV2: tela de login



**Figura 23 DV2 tela de login**

#### 4.2.7.2. DIAGRAMA DE SEQUENCIA: LOGIN

Conforme demonstrado no apêndice O

#### 4.2.8. CASO DE USO: VISUALIZAR QUESTIONÁRIOS



**Ator:** Administrador.

**Visão geral:** Este caso de uso tem por objetivo permitir ao administrador do sistema visualizar os questionários e suas respostas.

**Estado:** Concreto

**Pré-Condições:**

O usuário possuir login e senha de administrador do sistema.

- O usuário ao acessar a página inicial do sistema pressiona o link “Formulário” no menu.
- O sistema redireciona para a interface (DV3) que contém todos os formulários que foram cadastrados pelo administrador do sistema.
- Quando o usuário posicionar o mouse em cima de determinado formulário o ícone “questionários”  questionários aparecerá.
- Quando o usuário pressionar o ícone “questionários”, o sistema abrirá uma tela “Questionários” com a lista de questionários, o usuário poderá verificar a data e hora do preenchimento e visualizar as respostas.
- O usuário seleciona um questionário na lista e quando posicionar o mouse sobre o registro o ícone “visualizar respostas”  visualizar respostas aparecerá.
- Quando o usuário pressionar o ícone “visualizar respostas” o sistema exibirá a tela “Respostas do Questionário”.

**Cenários Secundários:**


- Caso não tenha sido criado nenhum formulário não será possível acessar esta funcionalidade.

## Fluxo Principal

P1. Visualizar Questionários

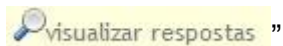
P1.1 Na página inicial do sistema o usuário pressiona o link “Formulários”.

P1.2 O sistema abre a tela (DV3).

P1.3 O usuário escolhe um formulário e pressiona o ícone “questionários”.

P1.4 O sistema exibe a tela “Questionários”.

P1.5 O usuário seleciona o questionário e pressiona o ícone “



P1.6 O sistema exibe a tela “Respostas do Questionários”.

## Fluxos Alternativos

Não há fluxos alternativos.

## Fluxos de Exceção

## Pós-condições

nenhuma.

## Requisitos Não Funcionais

- O usuário deve estar logado ao sistema.

### 4.2.8.1. TELAS

DV10: Questionários

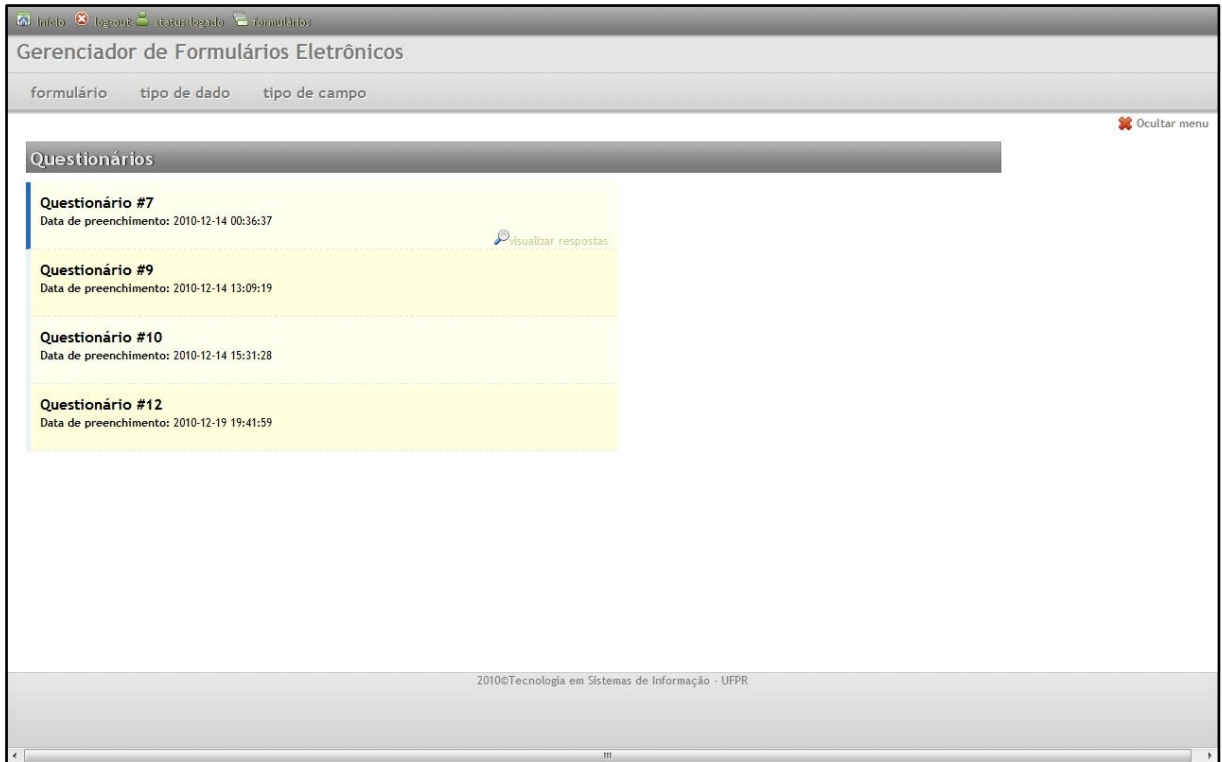


Figura 24 DV10 questionários

## DV11: Respostas do Questionário

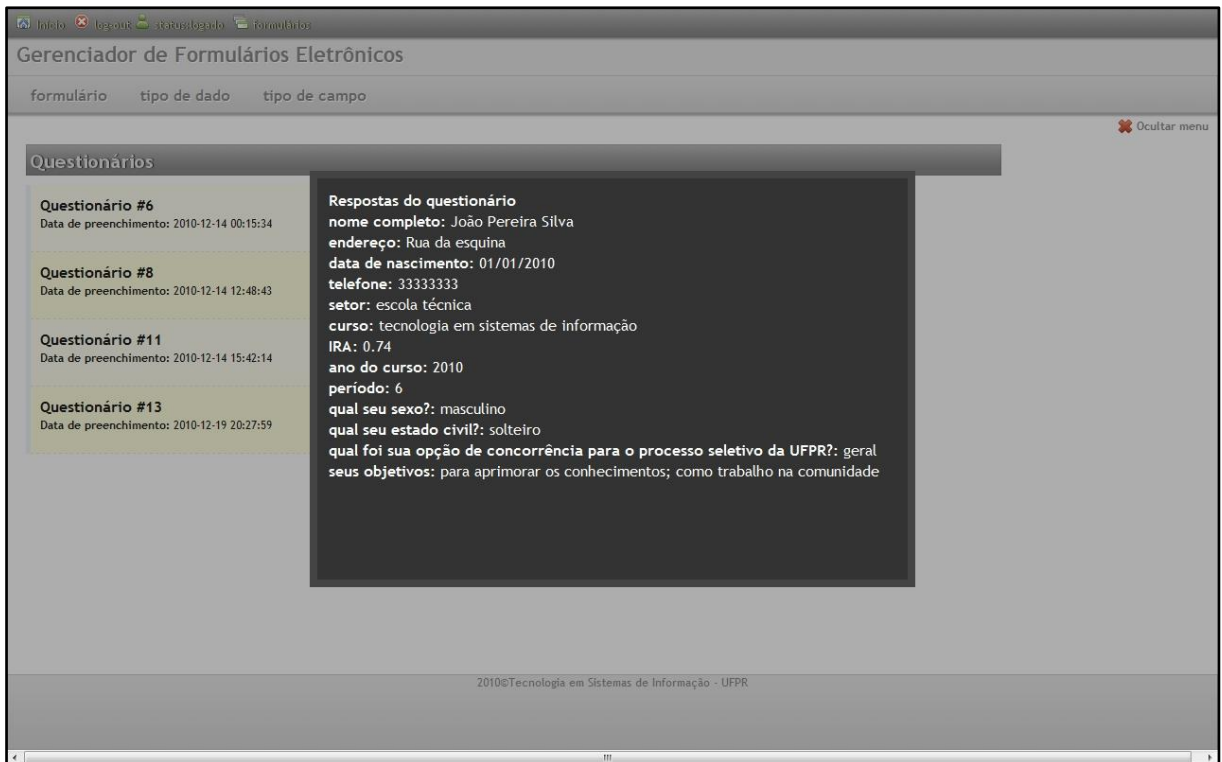


Figura 25 DV11 respostas do questionários

#### 4.2.8.2. DIAGRAMA DE SEQUENCIA: LOGIN

Conforme demonstrado no apêndice O.

#### 4.3. DIAGRAMA DE CLASSES

Conforme demonstrado no apêndice P.

#### 4.4. BANCO DE DADOS

##### 4.4.1. DIAGRAMA DE ENTIDADE-RELACIONAMENTO

Conforme demonstrado no apêndice Q.

##### 4.4.2. DICIONÁRIO DE DADOS

Conforme demonstrado no apêndice R.

#### 4.5. DEPENDÊNCIAS DO SISTEMA

Para seu funcionamento, o sistema Gerenciador de Formulários Eletrônicos depende de um navegador(*browser*) com javaScript habilitado, um servidor Apache e um banco de dados MySQL.

### 5. CONSIDERAÇÕES FINAIS

O Trabalho presente teve por objetivo a construção de um sistema para gerenciamento de formulários do departamento TADS da UFPR.



Até o presente momento a coleta de dados era feita de forma manual e descentralizada, o que ocasionava em problemas durante o gerenciamento da informação e a manutenibilidade das mesmas.

A idéia da construção de um sistema gerenciador de formulários eletrônicos, surgiu da proposta de informatizar o processo pelo qual se obtém as informações. Uma vez que o sistema é disponibilizado na internet e qualquer usuário pode ter acesso de qualquer computador que esteja conectado à rede mundial.

Durante o desenvolvimento do projeto buscou-se focar nas funcionalidades principais para que os objetivos fossem atingidos de maneira eficaz, de modo que, os problemas do mundo real fossem delineados para o mundo computacional.

Optou-se escolher ferramentas e tecnologias livres que auxiliassem no processo de modelagem e desenvolvimento do software para que as boas práticas e padrões fossem empregados garantindo a eficiência do sistema.

Pelo escasso tempo que tivemos, não foi possível ir além dos requisitos levantados, como por exemplo: Implementar a administração de usuários. O que seria interessante por proporcionar melhor gerenciamento nos formulários e questionários. Porém a forma como foi desenvolvido o sistema torna possível futuras implementações.

A partir deste projeto, tem-se uma nova concepção de coleta de dados de forma dinâmica, que vem a centralizar as informações e ajudar na dinâmica do processo.

## 5.1. IMPLEMENTAÇÕES FUTURAS

Cadastro de usuário para executar a administração do sistema;

Elaboração de relatórios gerenciais;

Implementação de mecanismo de consulta de questionários;

Paginação da listagem dos registros cadastrados.

## REFERÊNCIAS

- [BAL 94] BALSEMÃO, Luciana de Oliveira. HYPERPRO : uma ferramenta para a construção de hiperdocumentos no PROSOFT. Porto Alegre :CPGCC UFRGS, 1994.
- [WEI 92] WEISSHEIMER, Érico Olavo. Uma linguagem para criação de hiperdocumentos. Porto Alegre : CPGCC da UFRGS, 1992.
- [CON87] CONKLIN, Jeff. Hyperpept: an introduction and survey.Computer, Los Alamitos, v. 20, n.9, p. 17-21, 1987.
- CONVERSE,T.,PARK,J.PHP:a Bíblia.Rio de Janeiro:Campus,2003.Tradução da 3 edição.
- BENETT,Gordon.Intranets:Como implantar com Sucesso na sua empresa.Rio de Janeiro:Campus,1997.
- CURY, Antonio. Organização e métodos: uma visão holística. 8. ed. São Paulo: Atlas, 2005. [ISBN 85-224-4058-1](#)
- OLIVEIRA, Djalma. P.R, Sistemas, organização e métodos: uma abordagem gerencial. 15. ed. São Paulo: Atlas, 2005.
- CRANE,D.Ajax em Ação.1 Edição.São Paulo:Pearson Prentice Hall,2007.
- SEBESTA,R.W. Concepts of programming languages.7 Ed.Boston:Pearson/Addison,2005.
- SPÍNDOLA, Rodrigo Oliveira. Artigo: Projeto de software utilizando UML.
- Análise Orientada a Objetos. Autor: Bruno Guimarães de Faria e Silva, 2004. Disponível em:[HTTP://www.rootlinux.com.br/documentos/analise.pdf](http://www.rootlinux.com.br/documentos/analise.pdf).
- Acesso em: 20/11/2010.
- BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivan. UML Guia do Usuário. Editora Campus, 2000
- Wazlawick, Raul S. Análise e Projeto de Sistemas de Informação Orientados a Objetos. Editora Elsevier, 2004.
- GUEDES, Gilleanes Thorwald Araújo. UML: Uma Abordagem Prática. 3. ed. São Paulo: Novatec, 2008.
- LARMAN, Craig. Utilizando UML e Padrões. 2ª Edição, Porto Alegre, Editora Bookman, 2004.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. *Design patterns: elements of reusable object-oriented software*. Massachusetts, US: Addison-Wesley, 1995.

## BIBLIOGRAFIA

GOODMAN,D.JavaScript-Bible.5<sup>th</sup> Ed.Indiana:Wiley Publishing,2004.

Manuais da Linguagem de Programação PHP através do endereço eletrônico:  
<http://www.php.net/docs.php>.

Manuais do banco de dados MySQL através do endereço eletrônico:  
<http://www.dev.mysql.com/doc/Refman/5.1/en/>.

Welling,L.PHP e MySQL:desenvolvimento Web.3-ed.Rio de Janeiro:Elsevier,2005.

ACHOUR, Mehdi et al. **Manual do PHP**. Disponível em:  
<[http://www.php.net/manual/pt\\_BR/](http://www.php.net/manual/pt_BR/)>. Acesso em: 10 novembro 2010.

Jquery API Browser. Disponível em: <http://jquery.bassistance.de/apibrowser/> Último acesso em Novembro 2010.

## APÊNDICES

## APÊNDICE A – DIAGRAMA DE CASO DE USO

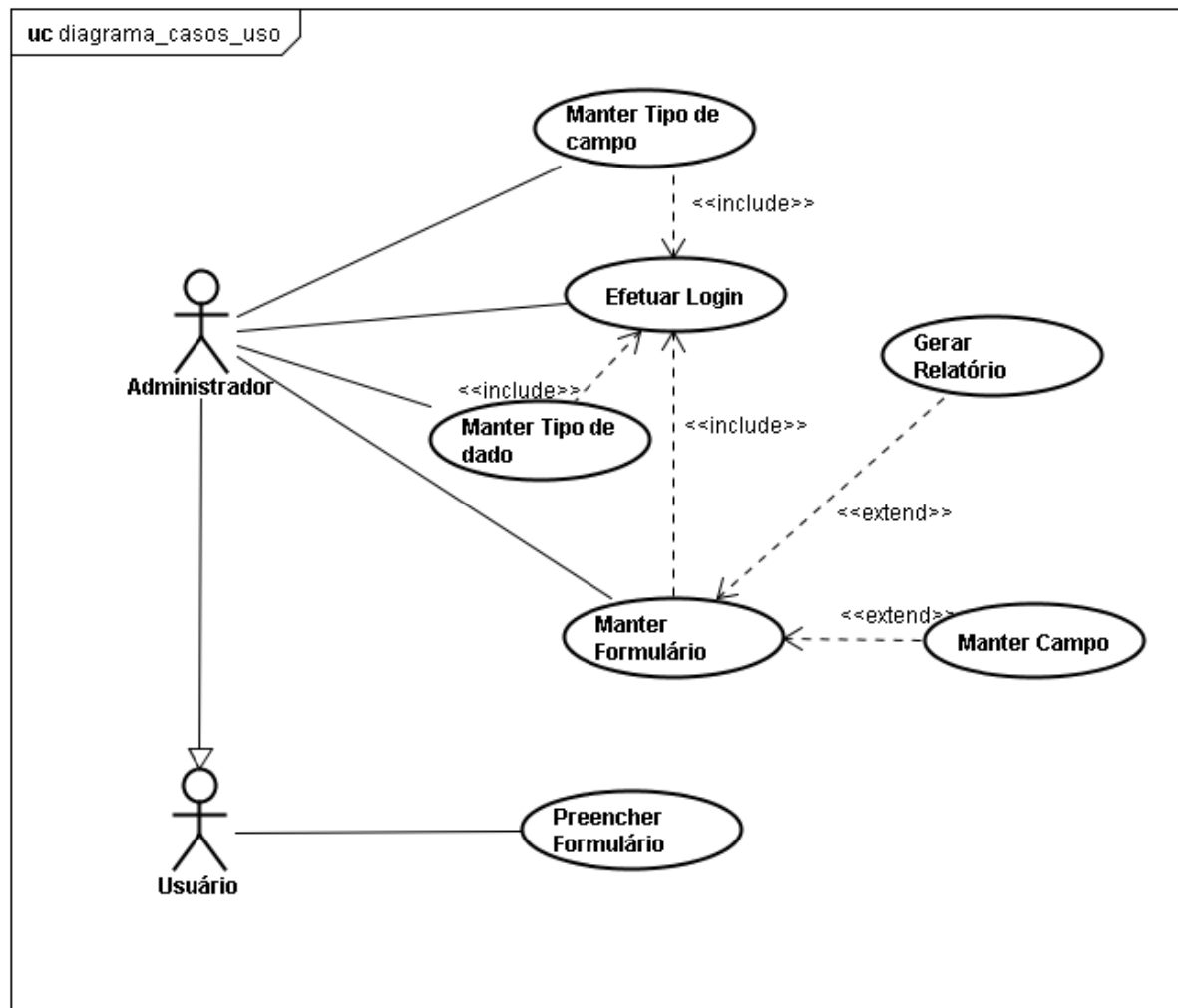


Figura 26 Diagrama de caso de uso

APÊNDICE B – DIAGRAMA DE SEQUÊNCIA: INCLUIR FORMULÁRIO

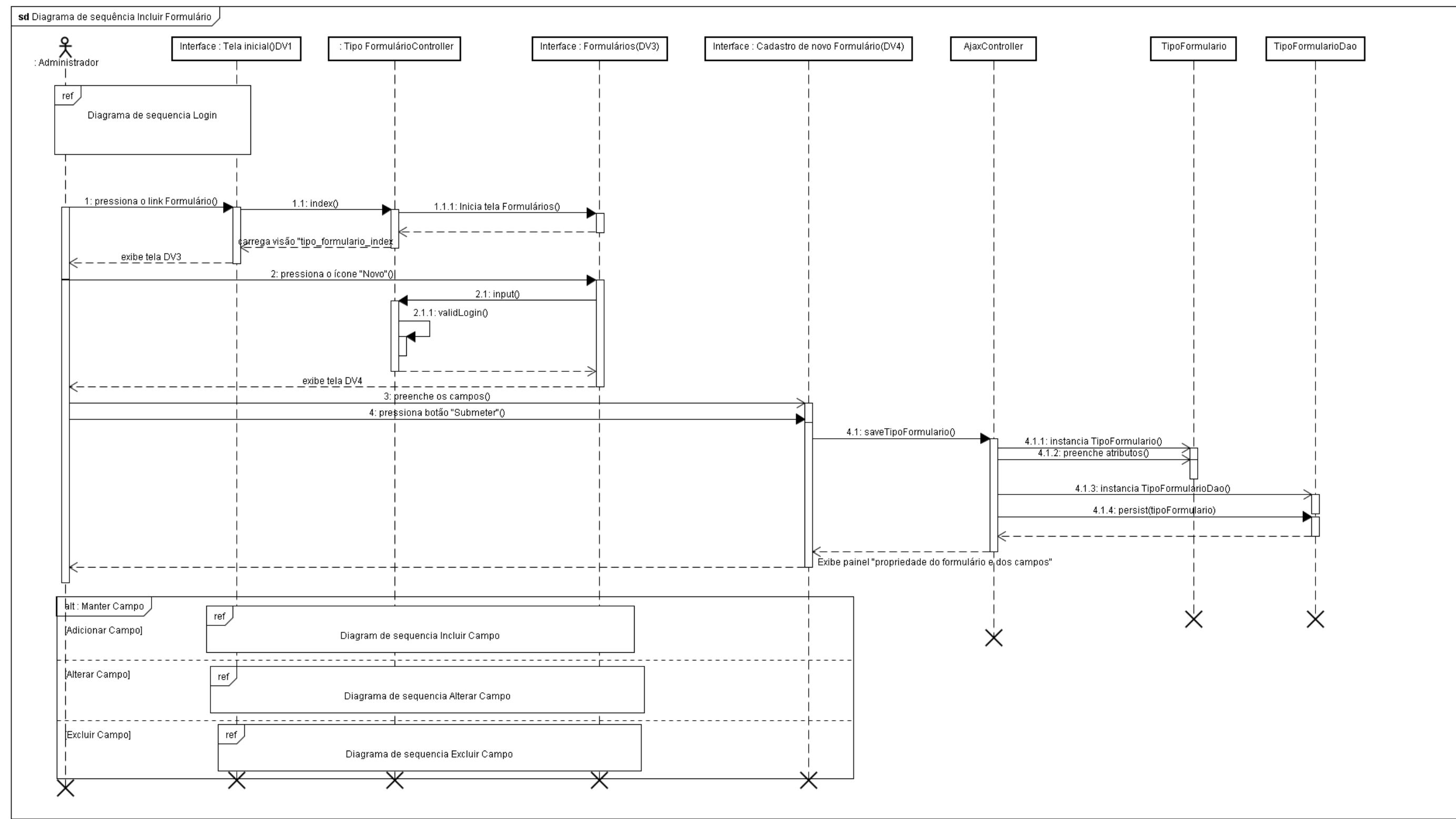
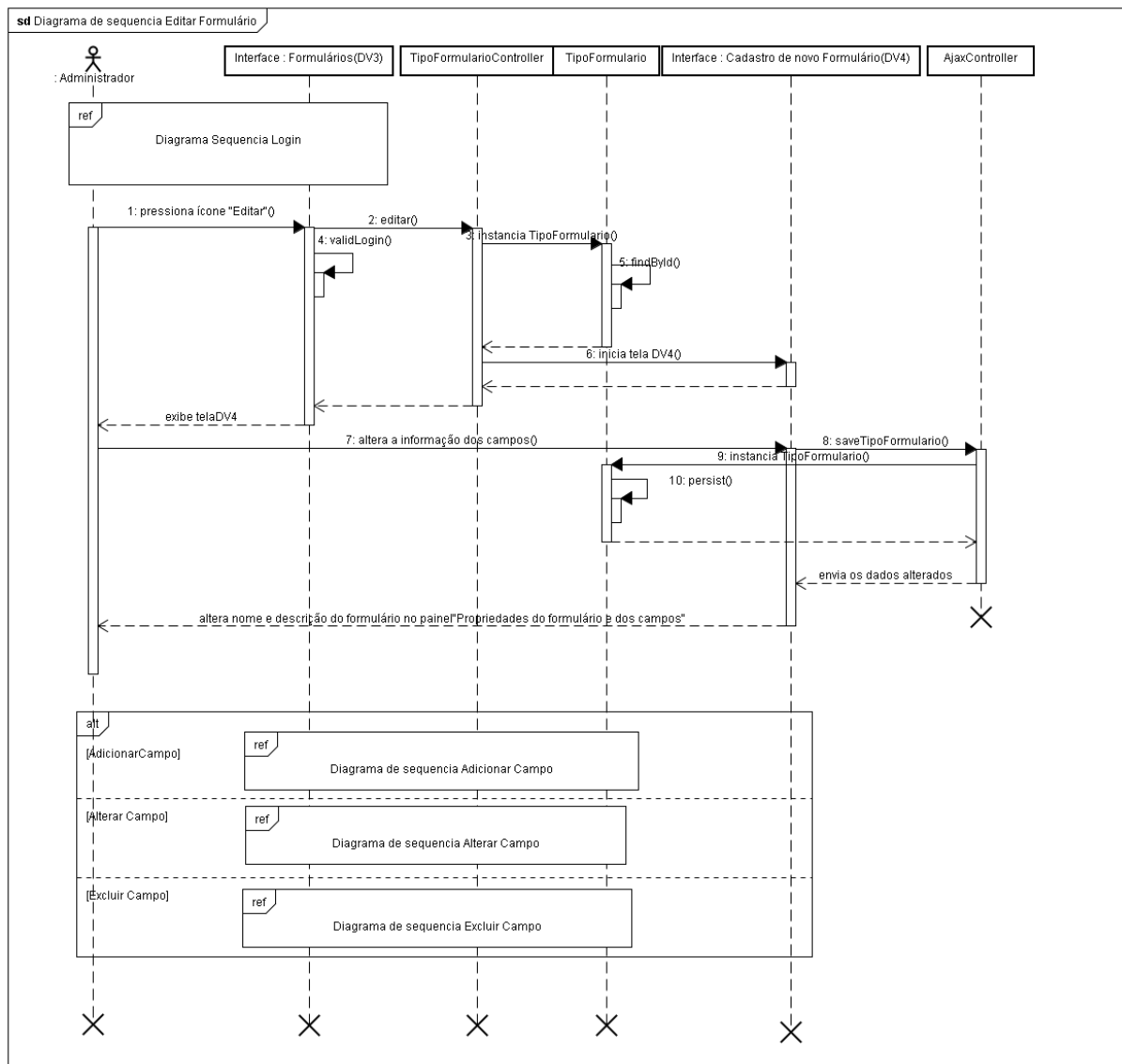


Figura 27 diagrama de sequência: incluir formulário

## APÊNDICE C – DIAGRAMA DE SEQUÊNCIA: ALTERAR FORMULÁRIO



**Figura 28 diagrama de sequência: alterar formulário**



## APÊNDICE D – DIAGRAMA DE SEQUÊNCIA: INCLUIR CAMPO

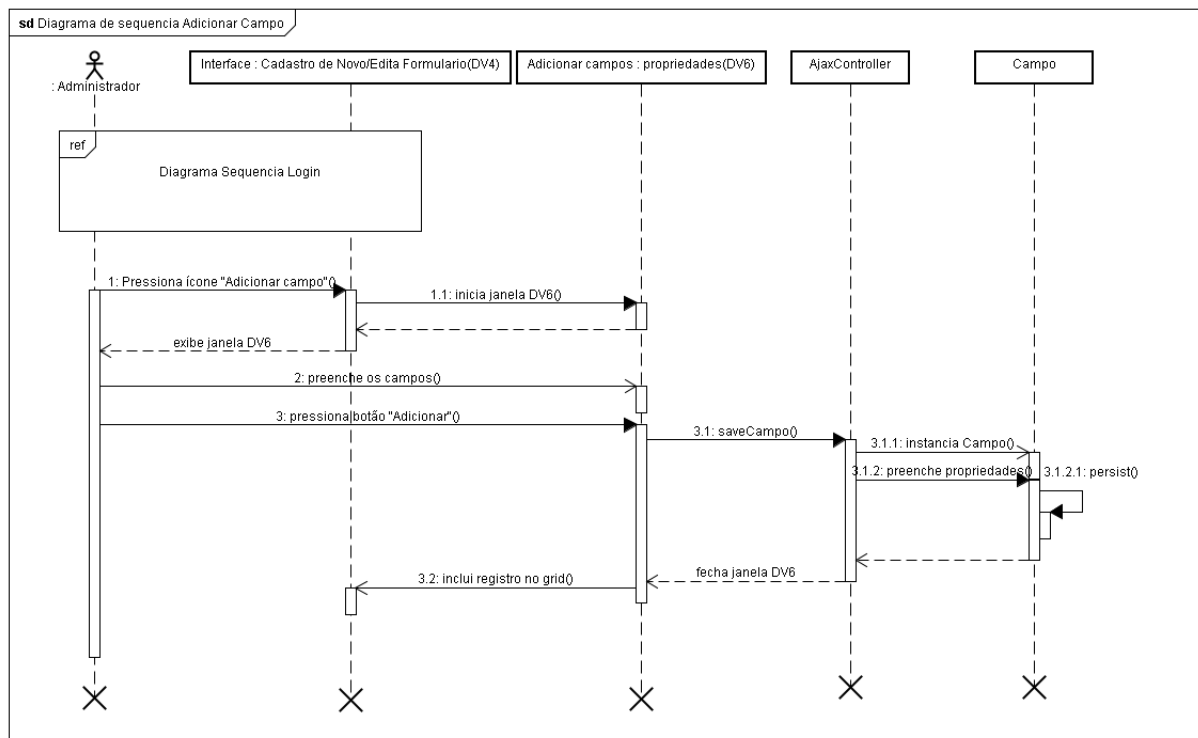


Figura 29 diagrama de sequencia: incluir campo

## APÊNDICE E – DIAGRAMA DE SEQUÊNCIA: ALTERAR CAMPO

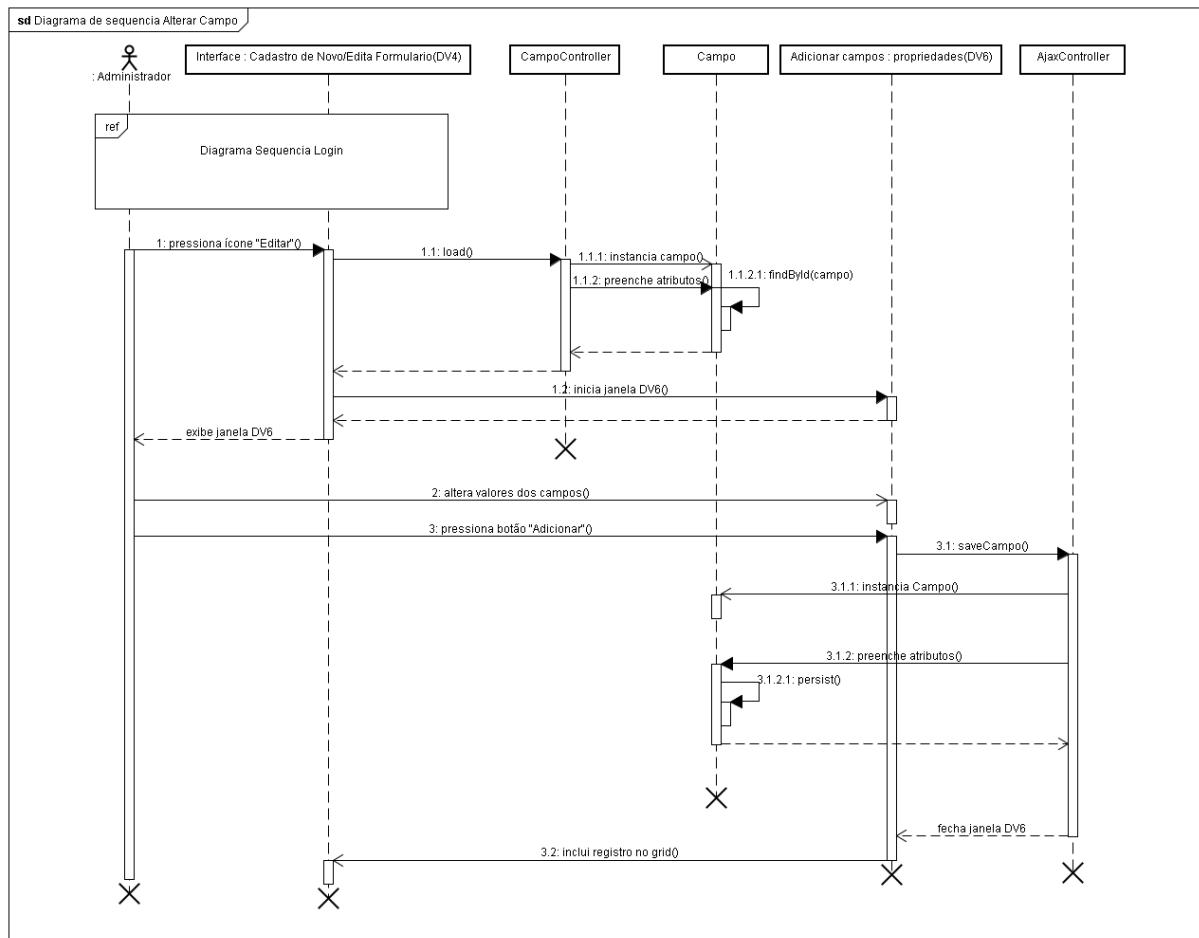


Figura 30 diagrama de sequência: alterar campo

## APÊNDICE F – DIAGRAMA DE SEQUÊNCIA: EXCLUIR CAMPO

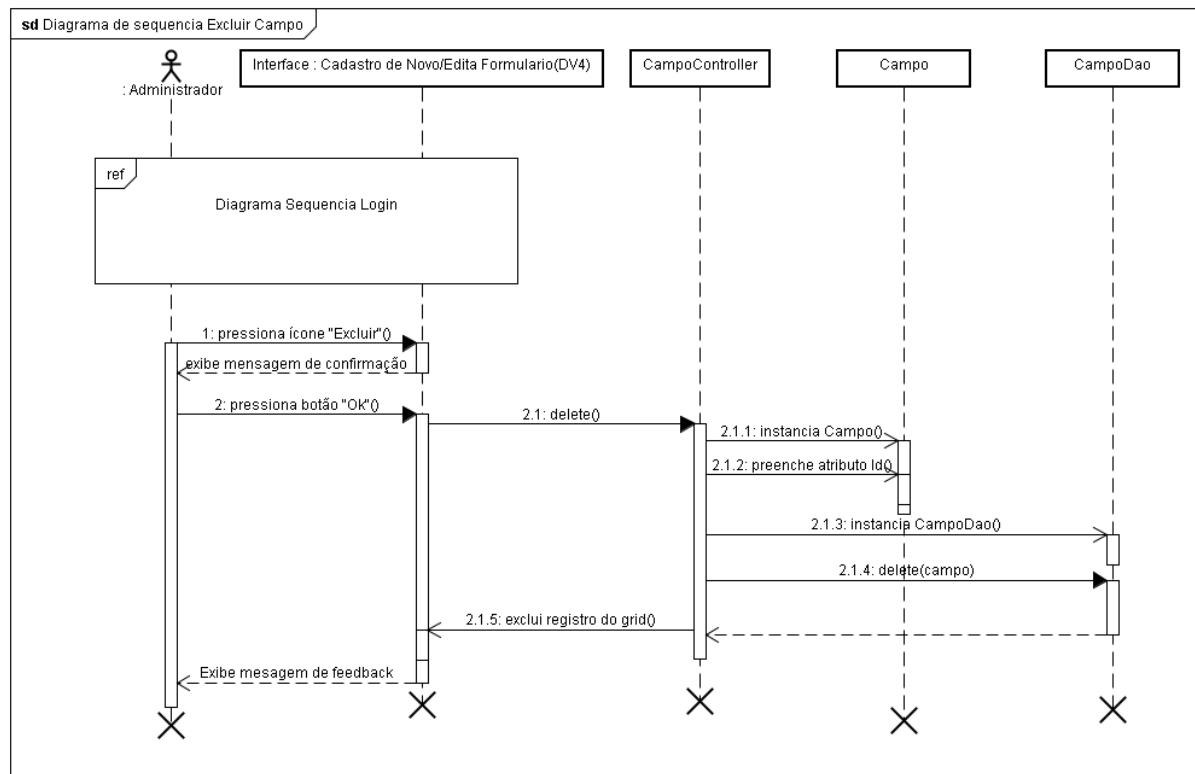


Figura 31 diagrama de sequência: excluir campo

## APÊNDICE G – DIAGRAMA DE SEQUÊNCIA: GERAR RELATÓRIO

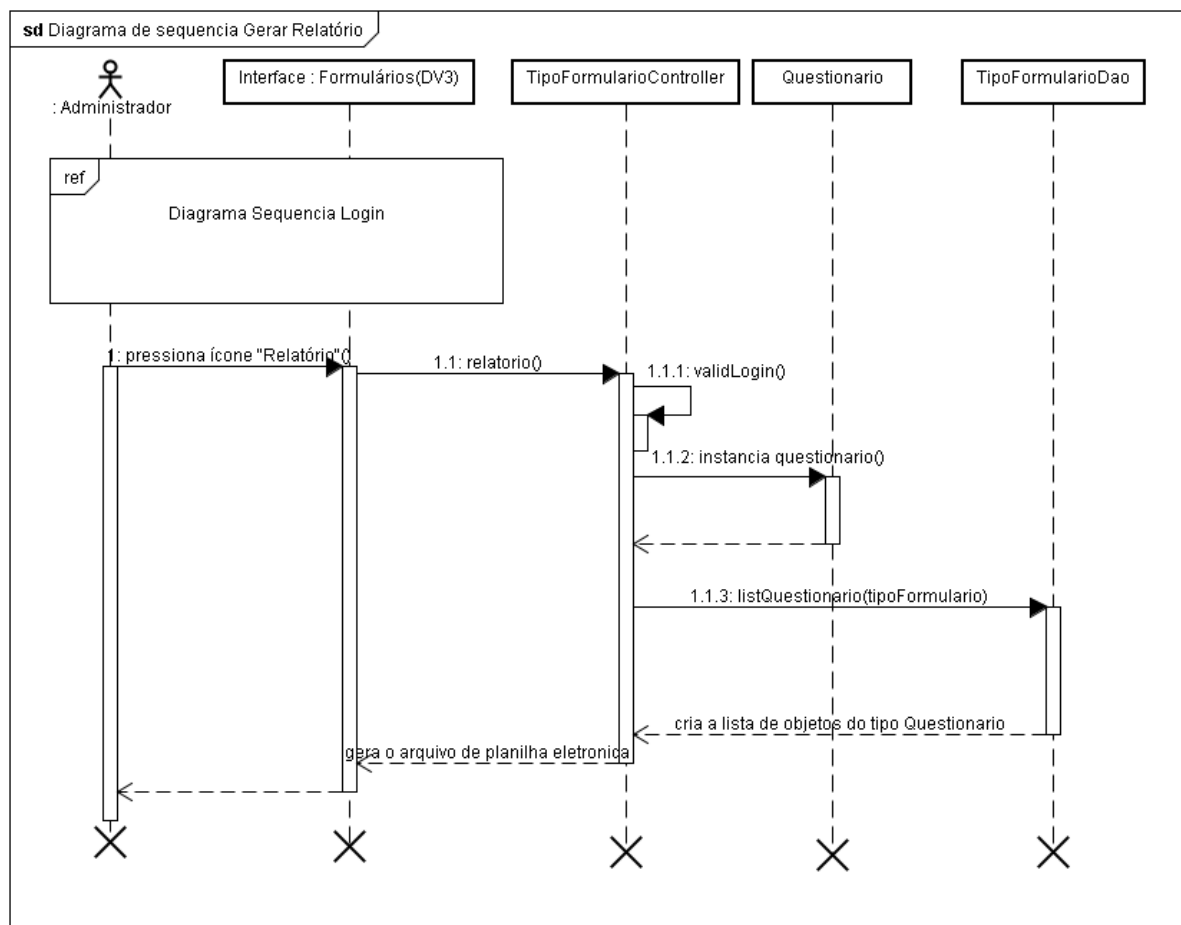
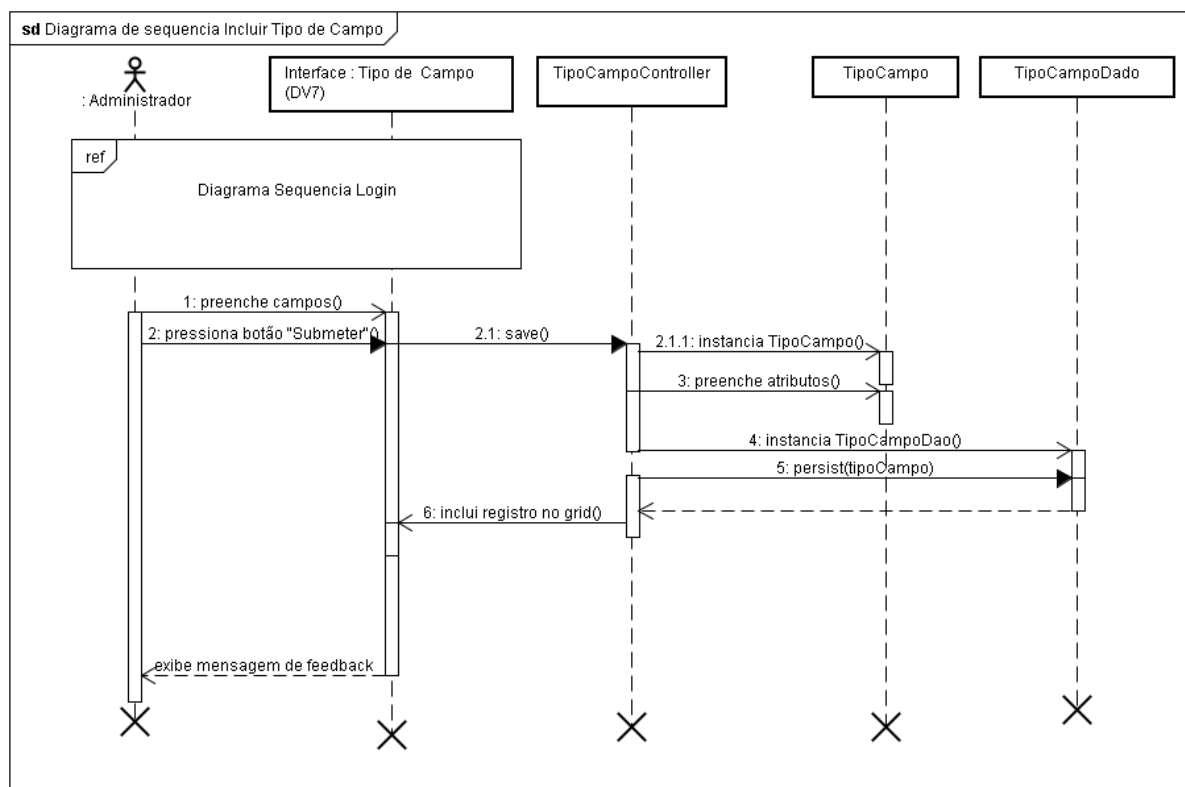


Figura 32 diagrama de sequência: gerar relatório

## APÊNDICE H – DIAGRAMA DE SEQUÊNCIA: INCLUIR TIPO DE CAMPO



**Figura 33** diagrama de sequência: incluir tipo de campo

## APÊNDICE I – DIAGRAMA DE SEQUÊNCIA: ALTERAR TIPO DE CAMPO

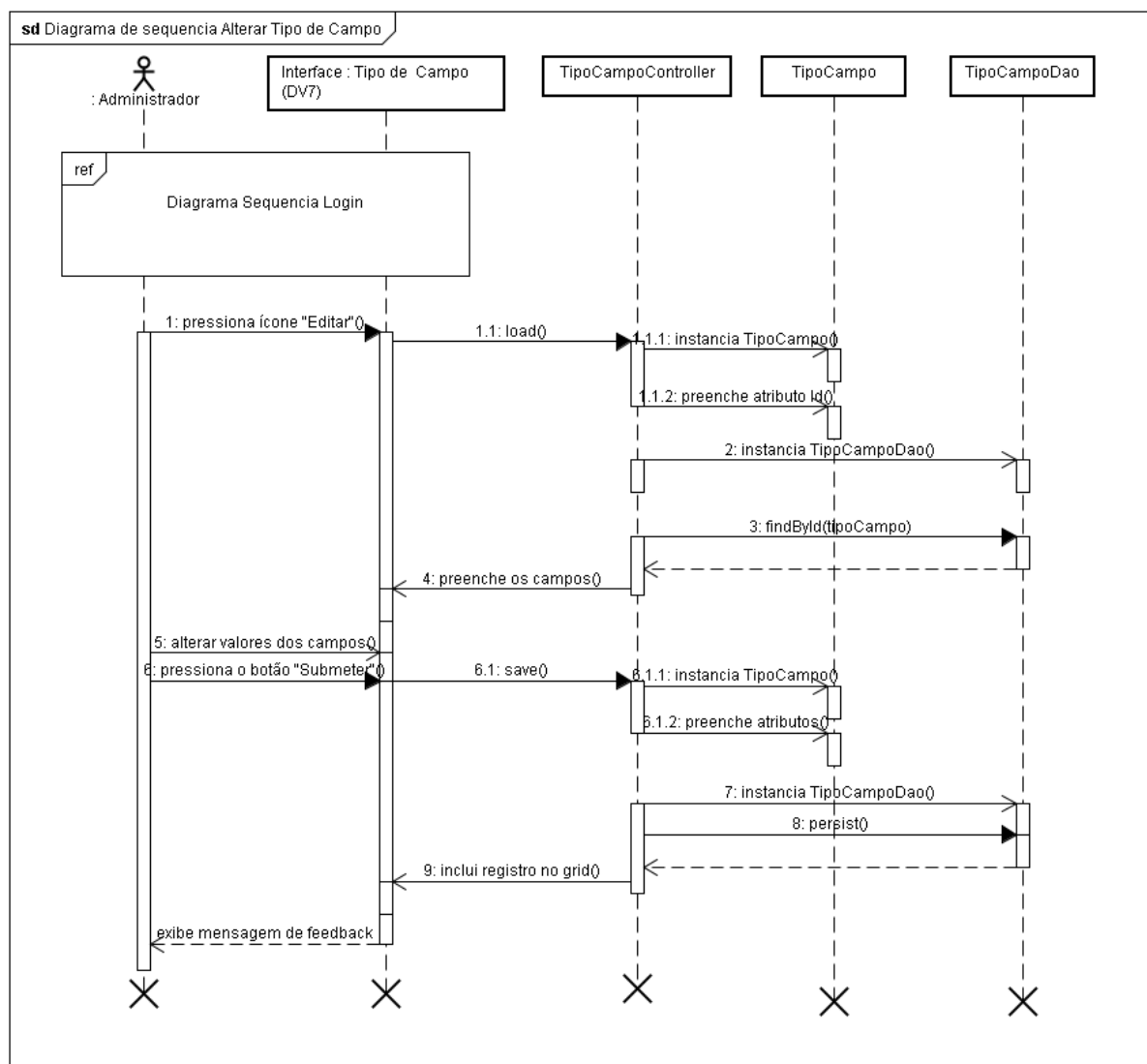


Figura 34 diagrama de sequência alterar tipo de campo

## APÊNDICE J – DIAGRAMA DE SEQUÊNCIA: EXCLUIR TIPO DE CAMPO

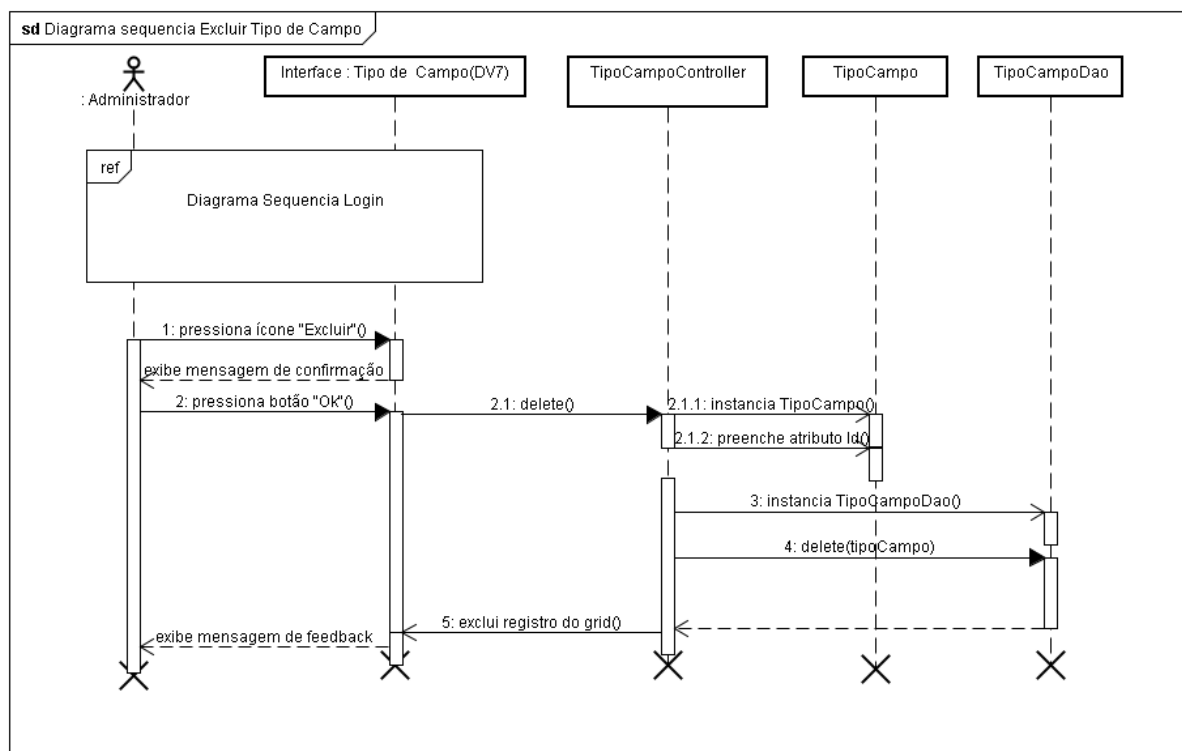


Figura 35 diagrama de sequência: excluir tipo de campo

## APÊNDICE K – DIAGRAMA DE SEQUÊNCIA: INCLUIR TIPO DE DADO

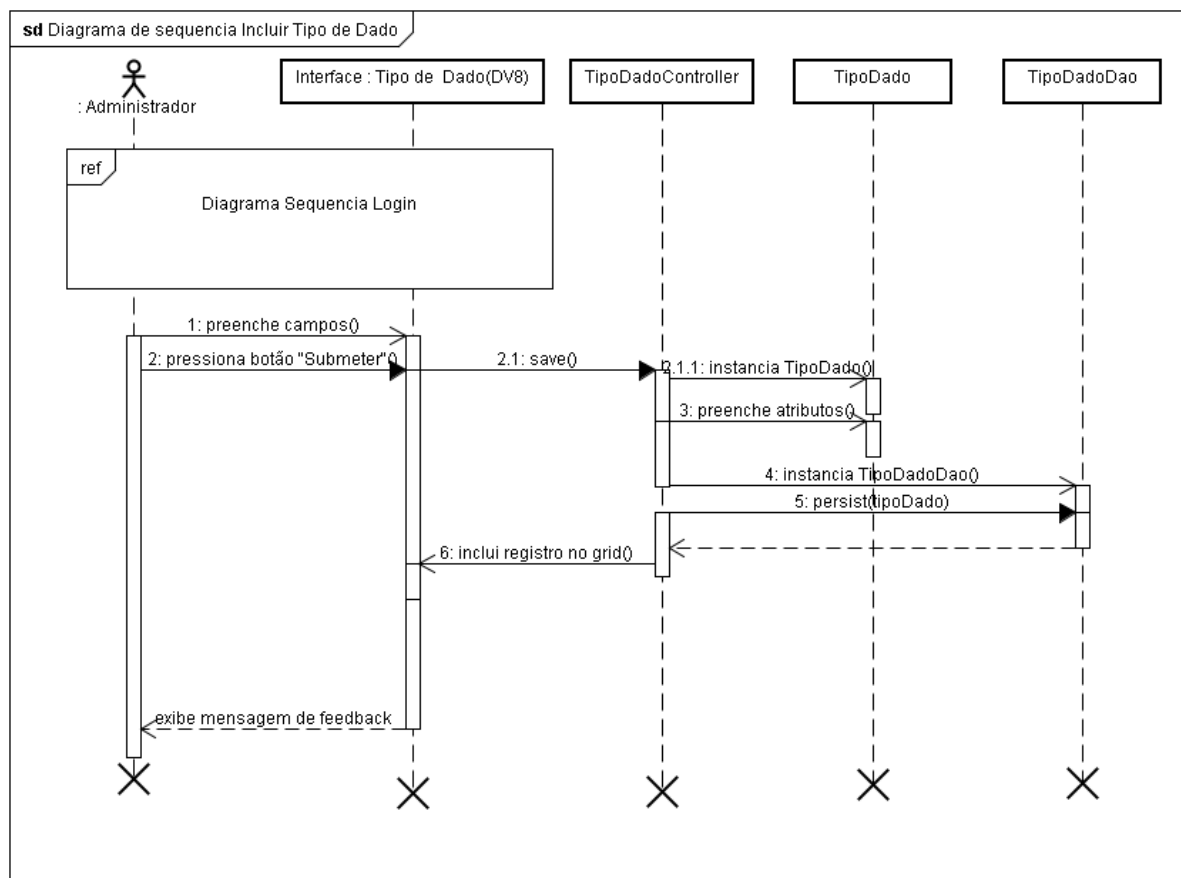


Figura 36 diagrama de sequência: incluir tipo de dado



## APÊNDICE L – DIAGRAMA DE SEQUÊNCIA: ALTERAR TIPO DE DADO

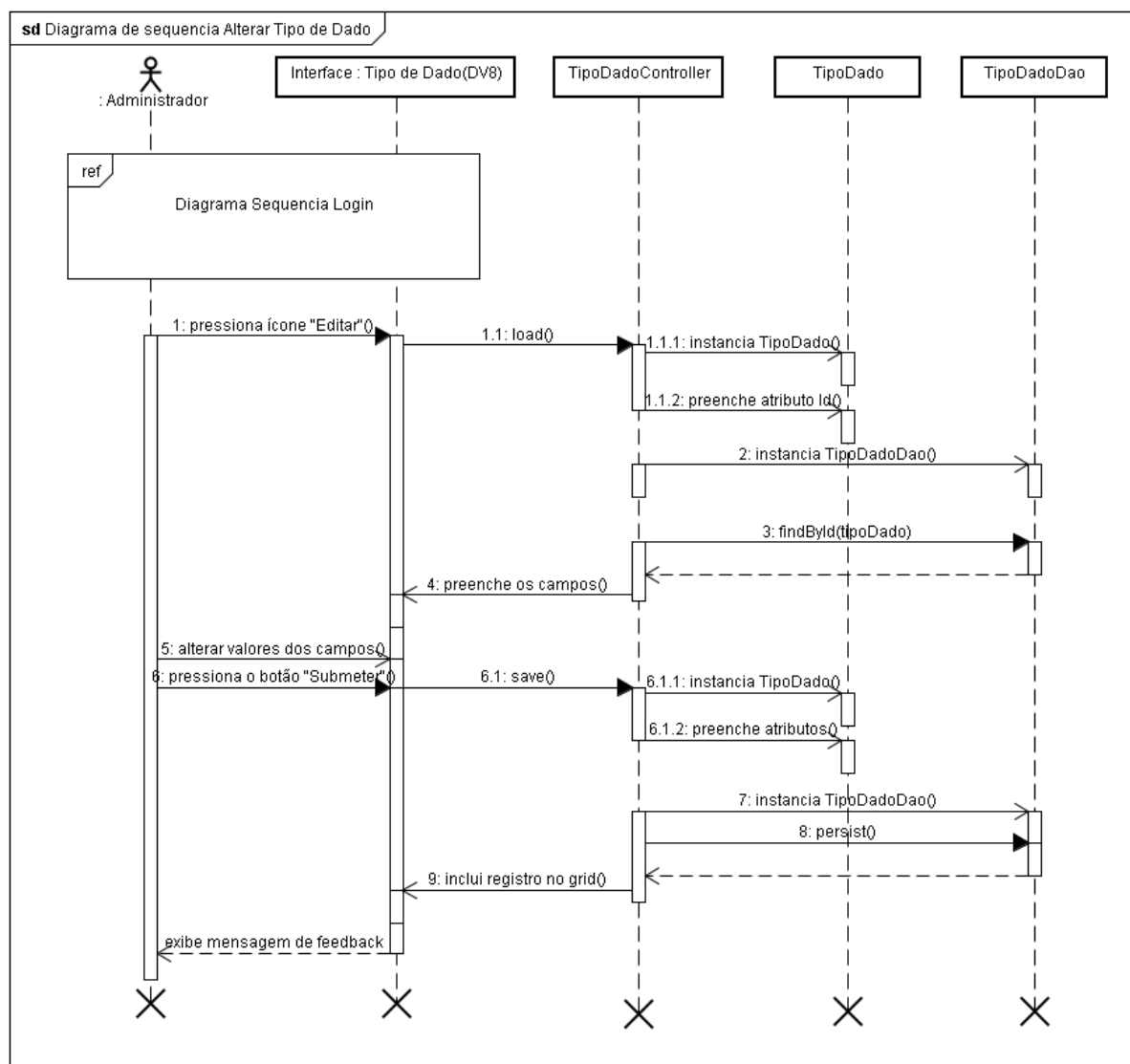


Figura 37 diagrama de sequência: alterar tipo de dado

## APÊNDICE M – DIAGRAMA DE SEQUÊNCIA: EXCLUIR TIPO DE DADO

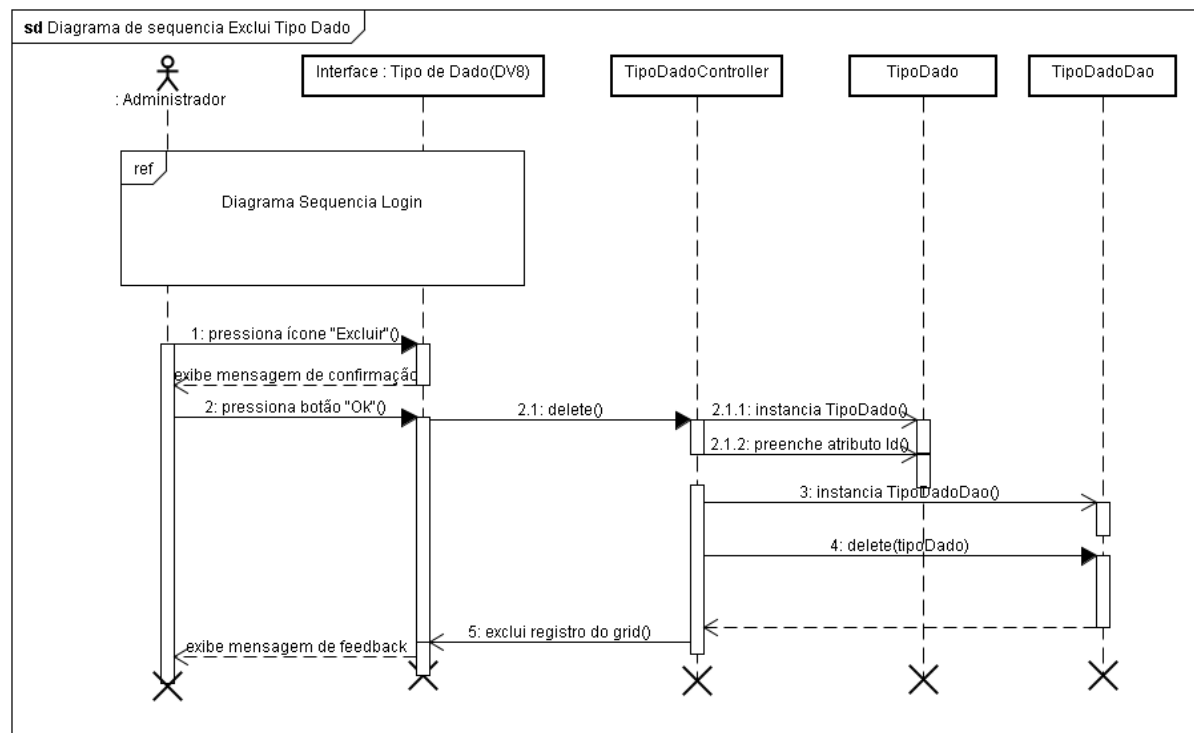


Figura 38 diagrama de sequência: excluir tipo de dado

# APÊNDICE N – DIAGRAMA DE SEQUÊNCIA: PREENCHER FORMULÁRIO

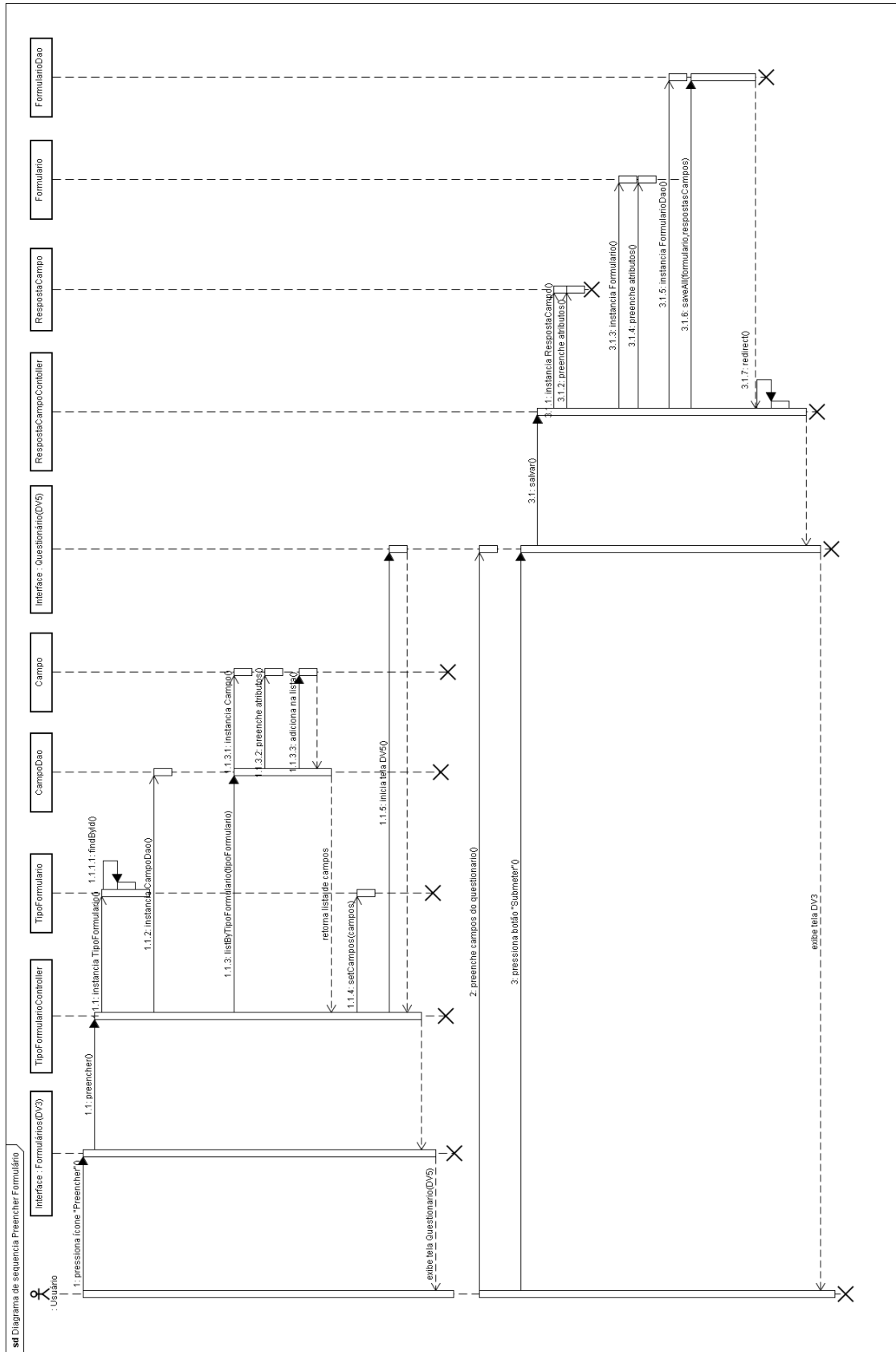


Figura 39 diagrama de sequência: preencher formulário

## APÊNDICE O – DIAGRAMA DE SEQUÊNCIA: LOGIN

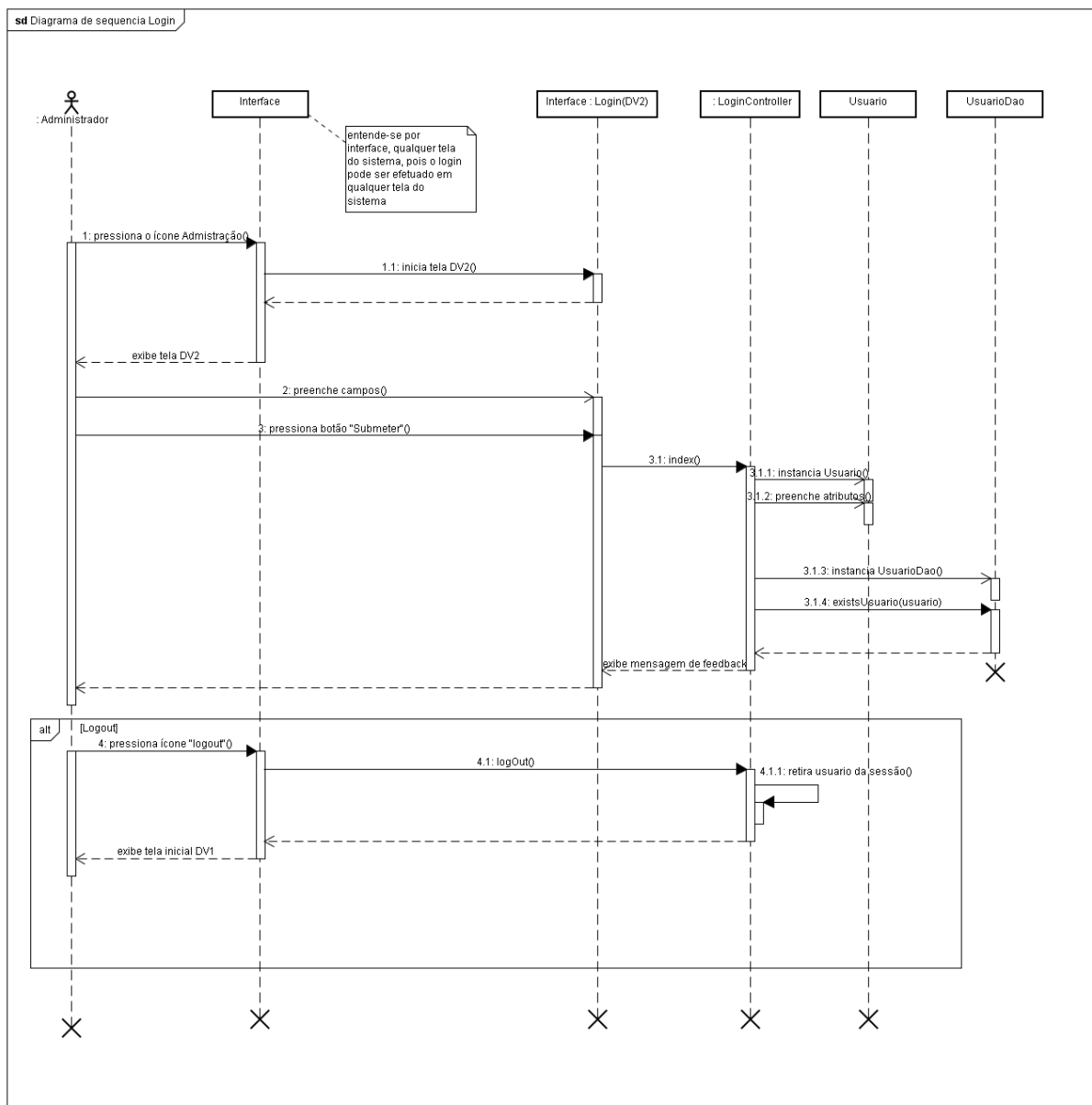
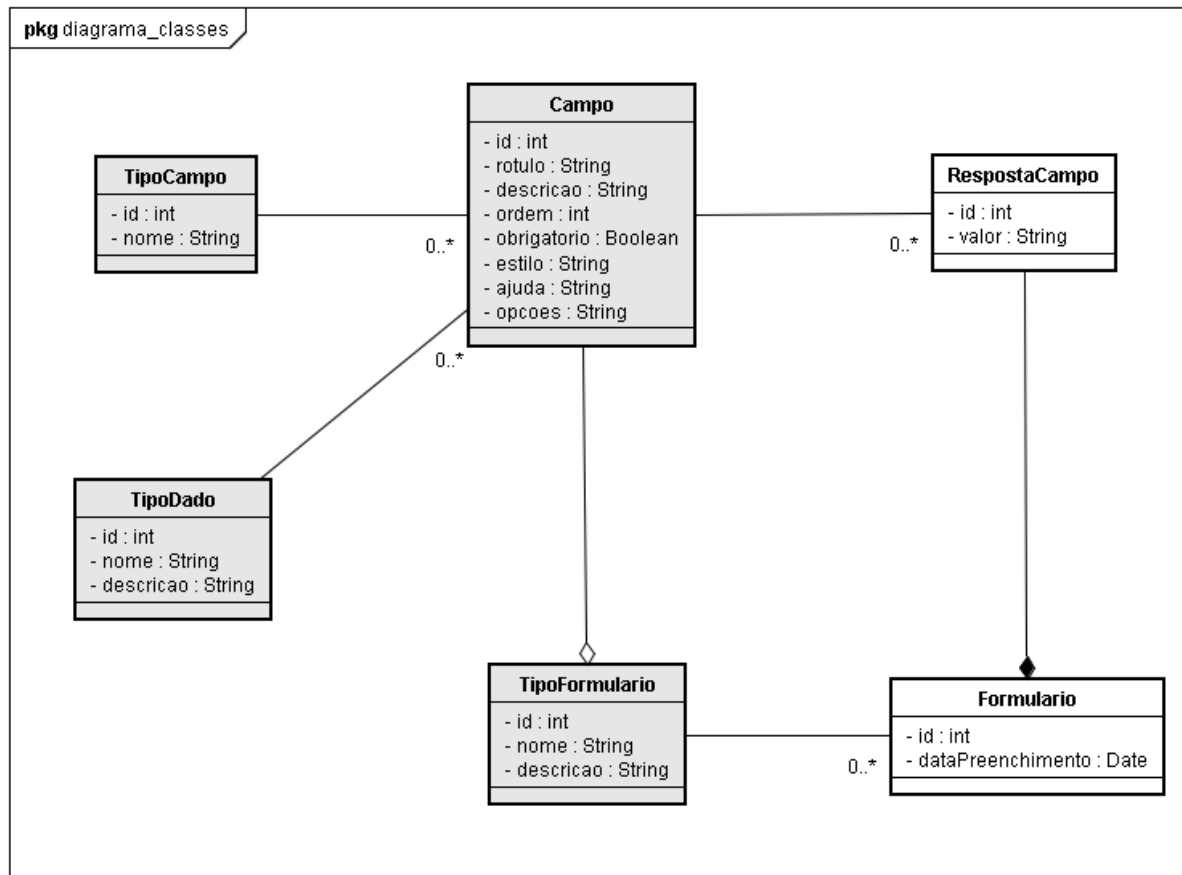


Figura 40 diagrama de sequência: login

**APÊNDICE P – DIAGRAMA DE CLASSES****Figura 41 diagrama de classes**

## APÊNDICE Q – DIAGRAMA ENTIDADE-RELACIONAMENTO

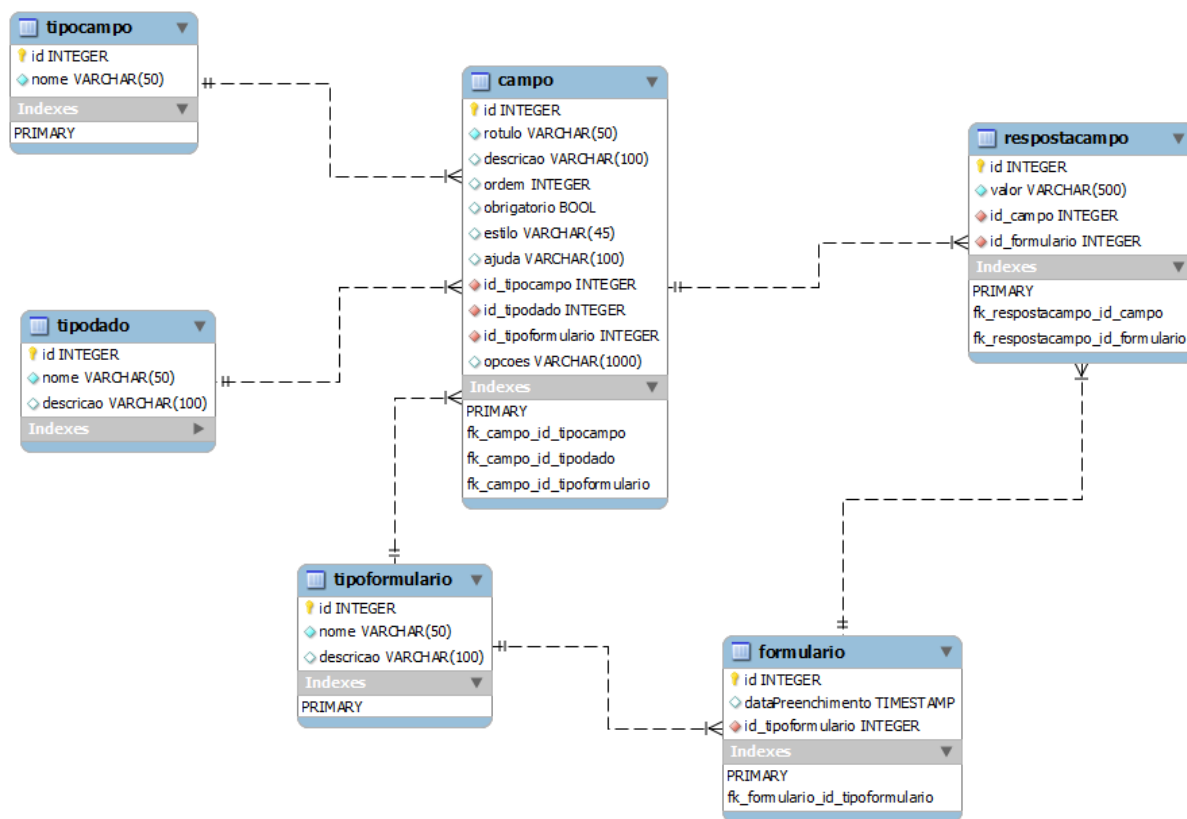


Figura 42 diagrama entidade-relacionamento

## APÊNDICE R – DICIONÁRIO DE DADOS

### campo

Campo	Tipo	Nulo	Padrão	Links para
<u>id</u>	int(11)	Não		
rotulo	varchar(50)	Não		
descricao	varchar(100)	Sim	NULL	
opcoes	varchar(1000)	Sim	NULL	
ordem	int(11)	Sim	NULL	
obrigatorio	tinyint(1)	Sim	NULL	
estilo	varchar(45)	Sim	NULL	
ajuda	varchar(100)	Sim	NULL	
id_tipocampo	int(11)	Não		tipocampo -> id
id_tipodado	int(11)	Não		tipodado -> id
id_tipoformulario	int(11)	Não		tipoformulario -> id

### formulario

Campo	Tipo	Nulo	Padrão	Links para
<u>id</u>	int(11)	Não		
dataPreenchimento	datetime	Sim	NULL	
id_tipoformulario	int(11)	Não		tipoformulario -> id

### respostacampo

Campo	Tipo	Nulo	Padrão	Links para
<u>id</u>	int(11)	Não		
valor	varchar(500)	Não		
id_campo	int(11)	Não		campo -> id
id_formulario	int(11)	Não		formulario -> id

### tipocampo

Campo	Tipo	Nulo	Padrão	Comentários
<u>id</u>	int(11)	Não		
nome	varchar(50)	Não		

### tipodado

Campo	Tipo	Nulo	Padrão	Comentários
<u>id</u>	int(11)	Não		
nome	varchar(50)	Não		
descricao	varchar(100)	Sim	NULL	

**tipoformulario**

Campo	Tipo	Nulo	Padrão	Comentários
<u>id</u>	int(11)	Não		
nome	varchar(50)	Não		
descricao	varchar(100)	Sim	<i>NULL</i>	

**usuario**

Campo	Tipo	Nulo	Padrão	Comentários
<u>id</u>	int(11)	Não		
login	varchar(20)	Não		
senha	varchar(20)	Não		



