

Resumo

Um dos grandes problemas enfrentados pelas organizações atualmente é a comunicação. Algumas soluções são adotadas, como e-mails e protocolos em papel, o que ajuda bastante, mas, muitas de suas formas de armazenamento, busca e organização são muito precárias. Os problemas podem ser percebidos ao se fazer solicitações ou reclamações no seu local de trabalho, em algum órgão ou alguma empresa na qual se comprou algum produto. Em geral é bastante difícil acompanhar o andamento de sua solicitação. Este projeto aborda um sistema para criação e acompanhamento de ocorrências, o qual tem como requisito ser genérico, tanto para adaptação à realidade da organização, quanto no trabalho diário. As tecnologias utilizadas visam tornar o sistema flexível, de fácil implantação e baixo custo. Assim, o sistema Help Desk Framework foi desenvolvido buscando o objetivo de ser um facilitador da informação.

Abstract

Communication is a considerable issue faced by organization nowadays. Some alternatives, such as e-mails and general documents in paper, are helpful, but the storage, search, and organization of these informations is too inadequate. This problem can be noticed when making requests or complaints in the workplace or in some store. In general, it is very difficult to track these requests/complaints. This project had developed an application to register and track requests, being adaptable both to the organization reality and to the day to day work. The used technologies were selected to allow the system flexibility, its easy deployment at a low cost. The Help Desk Framework system was developed to be an information facilitator.

Sumário

Índice de Figuras	vii
Índice de Tabelas	viii
Tabela de Símbolos e Siglas	ix
1 Introdução	11
1.1 Objetivos	12
1.2 Estrutura do Trabalho	12
1.3 Contribuições	12
2 Tecnologias Empregadas	13
2.1 Tecnologias no lado Servidor	13
2.2 Tecnologias no lado Cliente	15
2.3 XML	16
3 Modelagem da Solução	18
3.1 Organograma	18
3.2 Visão	20
3.3 Formulário	21
3.4 Estados	22
3.5 Usuário	22
3.6 Ocorrência	23
3.7 FAQ	24
4 Implementação	25
4.1 Estrutura	25
4.2 Ferramenta de Acompanhamento	27
4.3 Interface Externa	32
5 Conclusões e Trabalhos Futuros	35
5.1 Conclusões	35
5.2 Trabalhos futuros	35
Bibliografia	36
Apendice A Tabela de Variáveis da classe ConfigHTML	37
Apendice B Tabela de constantes do arquivo application.properties	38
Apendice C Documento de Especificação de Casos de Uso	39
Apendice D Diagrama de Classes	49

Índice de Figuras

Figura 1. Exame de aplicações para servidores WEB, entre Agosto de 1995 e Maio de 2007.	14
Figura 2. Etapas dos softwares necessários desde a solicitação do cliente até serem acessados os dados.	15
Figura 3. Comparativos de páginas sendo baixas com navegação convencional e Ajax.	17
Figura 4. Diagrama de classes relativa a Estrutura.	18
Figura 5. Exemplo do organograma da POLI.	19
Figura 6. Exemplo do organograma do Departamento de Sistemas Computacionais.	19
Figura 7. Diagrama de classes relativo a Visão.	20
Figura 8. Organograma e duas Visões para uma mesma organização.	20
Figura 9. Exemplo de um Formulário.	21
Figura 10. Exemplo de estados para dois Status.	22
Figura 11. Diagrama de classe relativo a Ocorrência.	24
Figura 12. Exemplo do uso das constantes do objeto ConfigHTML.	27
Figura 13. Página inicial da ferramenta de acompanhamento de Ocorrências.	28
Figura 14. Divisões das partes da página inicial.	29
Figura 15. Página para Edição de uma Ocorrência.	30
Figura 16. Locais nos arquivos de configurações para modificação no banco de dados.	31
Figura 17. Página para Edição de uma Ocorrência.	33

Índice de Tabelas

Tabela 1. Número de Sites Ativos para cada tipo de Servidor de aplicação HTTP no Início de Maio de 2007.	14
Tabela 2. Mapeamento dos objetos e propriedades, da Estrutura, referentes a Figura 5 e Figura 6.	26
Tabela 3. Funcionalidades do sistema e qual tipo de usuário pode utilizar.	29
Tabela 4. <i>Servlets</i> disponibilizado na interface disponibilizado externamente pelo Help Desk Framework.	32

Tabela de Símbolos e Siglas

- HTTP** *Hiper Text Tranference Protocol* (Protocolo de transferência de Hiper Texto).
- HTML** *Hiper Text Mark-up Language* (Linguagem de Marcação de Hiper Texto).
- WEB** Sinônimo de WWW que por sua vez significa Word Wide Web (A Grande Rede Mundial).
- XML** *eXtensible Markup Language* (Linguagem de Marcação Extensível).
- SGML** *Standard Generalized Markut Language* (Linguagem Padronizada de Marcação Genérica).
- SQL** *Structured Query Language* (Linguagem de Consulta Estruturada).

Agradecimentos

Quero iniciar esse agradecimento a uma força maior que nós, que alguns o chamam de Deus. Embora não acredite no Deus convencionado pelas instituições, mas sei que existe algo maior que nós e que nos faz recorrer quando precisamos. Abaixo dele tenho meus pais, Feijão e Betânia. Embora alguns atritos e muitas lutas conseguimos chegar aqui hoje e os amo muito. Mãe, um beijo.

A todos os meus irmãos (até agora seis), principalmente a Caio, Tibério e o pequeno João, que são os que tenho mais convivência a cada fase das suas vidas, pois têm em mim um irmão mais velho, e a quem eu tenho que ser exemplo.

Tive o prazer de conviver com meu avô José Taveira, e ainda tenho comigo minha avó Bui, que de alguma forma me ajudaram nessa caminhada. Não conheci em consciência meu avô José Vaqueiro, e cheguei depois da partida de minha avó Margarida, mas sei que também que me ajudaram muito, embora que em outro plano.

Meus tios e tias dos quais gosto bastante, com os quais sempre converso e que me dão bastante força. Em especial a meu tio Pedro, com quem tenho bastante amizade. Também todos os meus primos e primas. Principalmente a Blé e Georgina, que são minhas irmãs. Faço uma homenagem ao meu primo Renato, que nos deixou há alguns meses atrás.

Agora é a vez de agradecer aos meus amigos de Carpina (e região) por todos os momentos que tivemos e que construiu quem sou hoje. Aos que estão comigo sempre tendo dá força. Aos que foram torço por eles onde estiverem.

A uma equipe maravilhosa que conheci no C.E.S.A.R., empresa onde trabalho há dois anos e meio e que todo dia tenho uma nova lição. Jonas, Daniel, Vitor Braga, Rafael e todos os outros.

Meus professores que ajudaram no meu conhecimento. Em especial a Sérgio Soares, Carlos Alexandre, Márcio Cornélio, Fernando Buarque. Também agradeço a todos os outros. Alguns nos ensinaram conhecimento acadêmico e servem exemplo. Outros servem como exemplo do que não devo ser, mas também valeu o aprendizado.

Deixando por último, para dá mais suspense, meus irmãos que conheci na POLI. Alguns colegas e outros AMIGOS. Marcos, Victor, Rafael, Robson, Erik, Filipe, Diogo. Esses são quem tenho bastante convivência acadêmica, profissional e pessoal. E todos os outros que conheci durante esses seis anos da minha formação de ENGENHEIRO DA COMPUTAÇÃO.

Capítulo 1

Introdução

Atualmente, um dos grandes problemas enfrentados pelas organizações é o de comunicação. Algumas soluções são adotadas, como e-mails e protocolos em papel, o que ajuda bastante, mas para essas soluções as formas de armazenamento, de busca e organização são muito precárias. Podemos citar outros problemas importantes como segurança, confiabilidade e controle do que está sendo realizado.

Quem faz alguma solicitação ou reclamação em seu trabalho, em algum órgão ou em alguma empresa na qual comprou um produto, sabe o quanto é difícil acompanhar o andamento de solicitação requisitada.

Podemos dar como exemplo o sistema de suporte aos clientes de uma empresa. Quando um usuário faz uma solicitação, é interessante para ele ter conhecimento do status da sua solicitação, bem como para o próprio suporte da empresa, que precisa ter uma organização das solicitações. Muitas vezes, para uma solicitação ser recusada, teria que esperar na fila por bastante tempo, quando poderia logo no início, ter um filtro para verificar sua relevância. No mercado existem alguns sistemas que tentam contornar esses problemas. Esses sistemas são uma caixa fechada, onde a organização tem que se adaptar ao que o sistema propõe, ou se for o caso, desenvolver um que se molde às suas necessidades, o que pode custar muito caro.

O termo *Help Desk*, vindo do inglês, refere-se a um serviço de apoio, suporte e resolução de problemas para usuários de uma empresa, podendo tratar questões relacionadas a tecnologias da informação, dúvidas, pedidos de reembolso, entre outros. Este serviço de apoio pode servir tanto para clientes internos, por exemplo, profissionais que cuidam da manutenção de equipamentos e instalações dentro da empresa, quanto para clientes externos, como no caso de prestação de serviços. O serviço de *Help Desk* pode ser realizado tanto pessoalmente quanto por telefone, e-mail, fórum e outros meios de comunicação.

Alguns autores definem sistemas *Help Desk* como sistemas “facilitadores de informação” [6]. Uma outra concepção, a *Isolacionista*¹, tende a tratar e enxergar um sistema *Help Desk* como um simples sistema de suporte técnico.

¹ *Isolacionista*: relativo ou pertencente ao isolacionismo que é uma doutrina que preconiza o isolamento de um país do cenário internacional, mediante recusa a formar alianças, assumir compromissos econômicos externos e assinar acordos bilaterais [10].

1.1 Objetivos

Este trabalho baseia-se em uma concepção moderna de sistemas *Help Desk*, onde a principal característica deste é ser facilitador de “quaisquer informações” ao usuário, não importando o que seja essa facilitação [6]. Nesta concepção, não importa o tipo de informação tratada, como também qual a tecnologia utilizada para desenvolvimento e modelagem do sistema.

O objetivo deste trabalho é a criação de um sistema de ajuda para o cadastro, controle, atendimento, andamento e retorno de ocorrências. Esse sistema será genérico, tanto para se moldar às necessidades das organizações, como também para o gerenciamento por parte do administrador do sistema. A meta é desenvolver uma solução que facilitará as trocas de informações, sendo flexível, abrangendo vários tipos de informações, e com alta portabilidade.

Também é de interesse que a criação de ocorrências, reclamações e dúvidas criadas anteriormente tenham uma solução mais rápida, sem a necessidade de novos registros, e cadastro das mesmas. Tais ocorrências irão compor uma consulta rápida para que o usuário possa resolver seu problema de maneira simples e eficiente. Esse cadastro e exibição são conhecidos como FAQ, que é um cadastro de possíveis ocorrências e suas soluções.

1.2 Estrutura do Trabalho

No Capítulo 2 é apresentada uma abordagem geral das tecnologias utilizadas no desenvolvimento do trabalho, descrevendo as principais ferramentas, tanto do lado cliente como do servidor..

No Capítulo 3 é descrita a modelagem da solução do problema que se pretende resolver. Existe uma analogia entre a funcionalidade do sistema e os objetos utilizados para esse mapeamento. Também são apresentados exemplos de diferentes aplicações do sistema, para que além do entendimento, também seja observado a flexibilidade e portabilidade do mesmo.

No Capítulo 4 é apresentada a aplicação da tecnologia para a modelagem apresentada no Capítulo 3. Neste capítulo também é descrita a implementação, configuração e utilização do sistema.

Por fim, no Capítulo 5 é apresentada a conclusão do trabalho e sugestões para trabalhos futuros.

1.3 Contribuições

As contribuições deste trabalho são:

- A criação de uma ferramenta genérica de abertura e tramitação de Ocorrências;
- A utilização de tecnologias amplamente difundidas e de fácil manutenção;
- O baixo custo de implantação da ferramenta.

Capítulo 2

Tecnologias Empregadas

Uma das grandes preocupações em projeto de aplicações Cliente-Servidor, é a infraestrutura. A sua definição é fundamental e envolve questões como a escolha do sistema operacional, arquitetura de *hardware*, linguagem de programação e outros fatores. A infraestrutura influencia desde a modelagem do sistema até o momento que este entra em operação.

Nas seções seguintes, são descritas as ferramentas que foram utilizadas neste trabalho, no intuito de se obter uma melhor portabilidade, manutenção, escalabilidade e um item bastante importante: o custo. Inicialmente é descrita a tecnologia no lado do servidor e logo após no lado do cliente. Por fim, é explanado o XML.

2.1 Tecnologias no lado Servidor

Para que um serviço possa ser acessado remotamente, deve haver uma ou mais máquinas disponibilizando o serviço solicitado. A máquina que disponibiliza esse serviço é chamada de servidor. Um dos serviços mais empregado atualmente é o serviço utilizado quando se navega em alguma página WEB, podendo ser usados direta ou indiretamente.

Para uma máquina ser um servidor HTTP, ela precisa de alguma aplicação servidora instalada, para prover esse serviço. Existem diversas aplicações para servidores HTTP, dentre eles o Apache [13], servidor da *Apache Software Foundation*. Baseada em uma pesquisa realizada pelo site NetCraft, onde é feito periodicamente um exame dos servidores de aplicações WEB [11], foi constatado que o Apache é o servidor mais utilizado no mundo. Em dados retirados recentemente (início de Maio de 2007) foi constatado que este servidor roda em 56% dos servidores de aplicações HTTP no mundo [12]. Na Figura 1 e na Tabela 1 é apresentado um comparativo do servidor de HTTP Apache com alguns de seus concorrentes. O servidor Apache pode ser instalado em diversos sistemas operacionais, como Unix e Windows, dando uma grande portabilidade para as aplicações que forem utilizar este servidor.

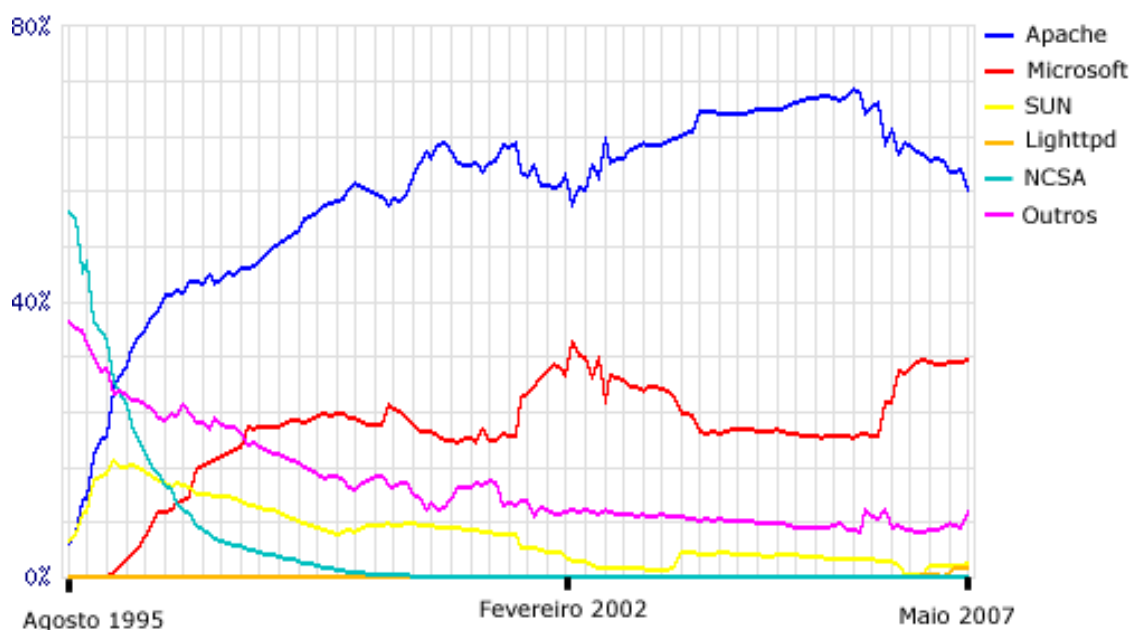


Figura 1. Exame de aplicações para servidores WEB, entre Agosto de 1995 e Maio de 2007 [12].

Tabela 1. Número de Sites Ativos para cada tipo de Servidor de aplicação HTTP no Início de Maio de 2007 [12].

Servidor	Sites Ativos	Porcentagem
Apache	66087698	56.00
Microsoft	37170290	31.49
Sun	2141252	1.81
Lighttpd	1411788	1.20

Contudo, como mencionado anteriormente, o servidor Apache é apenas servidor de HTTP. Por definição, páginas HTML são páginas estáticas. A idéia original desse serviço é o servidor mandar conteúdo no formato HTML e em geral, por outro lado, o que é enviado do lado do cliente são solicitações de outras páginas HTML. Com o crescimento da Internet foi aumentando a necessidade de uma maior interação entre o cliente e o servidor. Foi a partir desta necessidade de interação que surgiram as linguagens de processamento no lado do servidor. Neste caso o servidor conseguia processar e trabalhar com dados vindos do cliente, e não somente retornar páginas após serem pressionados alguns *links*.. Essa melhoria foi decisiva para o grande salto da Internet que existe atualmente.

Uma extensão do Apache para processamento de dados vindo do lado do cliente é o Apache Tomcat[1]. O Tomcat é um serviço responsável pela execução de programas escritos em Java [7]. Este serviço recebe do Apache a página solicitada juntamente com os argumentos vindos do lado cliente, realiza o processamento e devolve uma resposta em formato HTML para o Apache, que por sua vez envia para o cliente. Neste processamento ocorre todo o dinamismo das páginas. Por exemplo, quando um formulário de cadastro é preenchido (um pagamento em um site do banco, ou uma conversa em um bate-papo), a parte de processamento é a responsável pela sua interação.

Para completar essa interatividade, é necessário que os dados sejam guardados e posteriormente recuperados em bancos de dados. Um sistema gerenciador de banco de dados (SGBD) é quem organiza os dados, salvando-os e criando uma interface para o tratamento dos mesmos. Ele está em uma camada entre a aplicação que o usuário trabalha, e os dados propriamente ditos. Um SGBD muito utilizado atualmente é o MySQL [14]. Uma ferramenta com licença de baixo custo para aplicações comerciais (e sem custo para aplicações pessoais e aprendizado), fácil utilização e boa portabilidade, o que proporcionam um ótimo custo benefício.

Um modo de facilitar a comunicação da aplicação com o SGBD é o mapeamento dos dados do SGBD em objetos. A linguagem Java trabalha com objetos, e desta forma, no desenvolvimento da aplicação somente é necessário manipular os objetos e sem a utilização de comandos SQL. O Hibernate [4] é um *framework* que possibilita esse mapeamento.

Após a apresentação dos principais agentes no desenvolvimento de um sistema WEB, é necessário relacionar quais e como esses agentes foram utilizados no desenvolvimento deste trabalho. Na Figura 2 é apresentada cada etapa da relação entre os agentes, desde a solicitação de um cadastro, até a persistência de dados no MySQL.

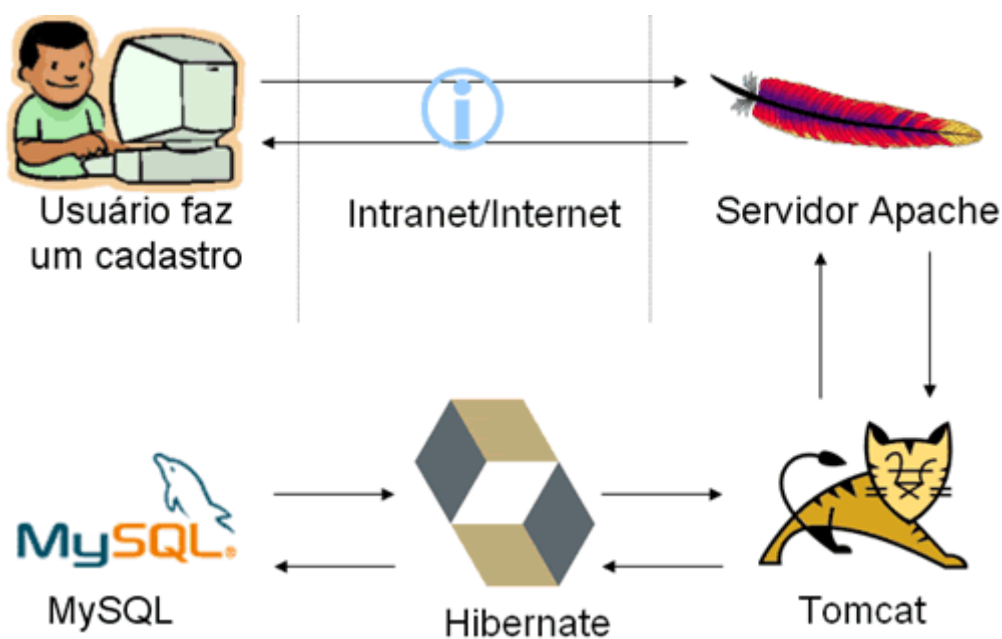


Figura 2. Etapas dos softwares necessários desde a solicitação do cliente até serem acessados os dados.

2.2 Tecnologias no lado Cliente

Navegadores Web são utilizados para navegar em páginas da Internet. Os serviços HTTP, que são utilizados na navegação, transferem dados em formato HTML como também imagens, vídeos e outros tipos de arquivos. Os navegadores Web mais comuns, como o Internet Explorer, Mozilla e Opera, implementam a versão 1.1 do HTTP. Então, sempre que um usuário acessa um site, o navegador Web recebe um texto HTML, que pode incluir caminhos para figuras, animações e scripts.

Alguns objetos dinâmicos, animações e validações de formulário não precisam ser enviadas ao servidor para ocorra alguma ação. Algumas tarefas podem ser realizadas ainda no navegador Web do cliente, não necessitando ser enviada ao servidor, o que diminui o tempo de resposta ao cliente e a carga de processamento no servidor. Esse dinamismo é obtido através de funções em scripts, que são chamadas a partir de algum evento no navegador Web, como por exemplo, abrir a página, clicar em um botão, passar o mouse sobre uma imagem e outros. A linguagem de script mais comum e difundida nos principais navegadores Web atualmente é o JavaScript [9].

Em uma página HTML convencional, sempre que se acessa um link ou submete-se um formulário, o navegador Web envia a requisição e baixa todo o conteúdo novamente. Um tempo considerável é gasto na transmissão de uma página cheia de conteúdo, figuras e animações, principalmente quando essa página está em um servidor de maior distância e com grande tráfego de acesso.

Uma evolução do modo de trabalhar com páginas HTML é o Ajax [2]. O Ajax não é uma ferramenta nova, nem mesmo uma atualização para os navegadores Web, mas sim um novo meio de navegação utilizando HTML e JavaScript. Através de alguns objetos nativos dos navegadores Web, junto com JavaScript, começa-se a fazer o envio e a atualização de partes das páginas que precisam ser atualizadas, e não de toda a página. Na Figura 3 é apresentado um exemplo de uma página simples que pode ser baixada completamente comparando-a com o Ajax. Na Figura 3 (a) é mostrada uma página inicial que quando o usuário clicar no nome, vai ser exibida uma mensagem. Na Figura 3 (b) é apresentado como seria uma página convencional a ser baixada, e na Figura 3 (c) é ilustrada a mesma mensagem que será baixada utilizando Ajax.

No exemplo apresentado na Figura 3 (b), que é a página sem Ajax, seria aberta uma nova página completa, e não seria utilizada a anterior. No segundo caso, a primeira página, Figura 3 (a), seria mantida e o conteúdo da *tag SPAN* seria inserido somente a mensagem “Bom Dia” que está em uma outra página, Figura 3 (c). Baseando-se neste exemplo, é possível imaginar a quantidade de dados que precisariam ser recarregados para um grande portal com vários textos, imagens e objetos.

Ajax não é uma nova tecnologia, mas uma nova abordagem para trabalhar com tecnologias já existentes como HTML, folhas de estilo (CSS), JavaScript e XML. Suas ramificações vêm desde o fim da década de 90. Atualmente vem ganhando bastante espaço e é o carro chefe da WEB 2.0 [8].

2.3 XML

Sendo um sub-tipo do SGML, XML é uma poderosa linguagem capaz de descrever diversos tipos de dados [5]. Seu propósito inicial é a transferência de informação na Internet por diversos meios. Os conteúdos em XML também podem ser usados para armazenar informação.

Criado pelo *World Wide Web Consortium (W3C)*, o XML baseou-se na idéia de criar uma linguagem de marcação flexível semelhante a SGML e com a simplicidade do HTML. O princípio do projeto era criar uma linguagem que pudesse ser lida por software e integrar-se com as demais linguagens. Vários fatores importantes são integrados a sua filosofia, como:

- Simplicidade e legibilidade para ler o conteúdo, tanto para humanos quanto para computadores;
- A possibilidade de criar inúmeras *tags* de diferentes finalidades e níveis;

- Possibilidade de separação do conteúdo e formatação visual;
- Integração com diversos tipos de software e banco de dados;
- Criação de arquivos para validação da estruturas, conhecidos como Documento de Definição de Tipo (*Document Type Definition - DTD*).

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2 <HTML>
3 <HEAD>
4 <TITLE>Help Desk Framework</TITLE>
5 <META NAME="Generator" CONTENT="EditPlus">
6 <META NAME="Keywords" CONTENT="">
7 <META NAME="Description" CONTENT="">
8 </HEAD>
9 <BODY>
10 <span onClick="abrirPagina()"> Clique aqui para ver a mensagem!! </span>
11 <span id="mensagem"> </span>
12 </BODY>
13 </HTML>
14 <script language="javascript">
15 function abrirPagina(){
16 // ... Outros códigos aqui
17 }
18 // ... Outros códigos aqui
19 </script>

```

(a). Uma página inicial que quando o usuário clicar, será exibida uma mensagem.

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2 <HTML>
3 <HEAD>
4 <TITLE>Help Desk Framework</TITLE>
5 <META NAME="Author" CONTENT="Júlio Taveira">
6 <META NAME="Keywords" CONTENT="">
7 <META NAME="Description" CONTENT="">
8 </HEAD>
9 <BODY>
10 <span id="mensagem"> Bom Dia </span>
11 </BODY>
12 </HTML>

```

(b). Uma página convencional mostrando a mensagem Bom Dia.

```

1 Bom Dia

```

(c). Uma página com Ajax contendo somente a mensagem que vai ser exibida.

Figura 3. Comparativos de páginas sendo baixas com navegação convencional e Ajax.

Capítulo 3

Modelagem da Solução

Como descrito na Seção 1.1, o objeto deste trabalho é a criação de um sistema para o cadastramento, controle, atendimento, andamento e retorno de ocorrências. O grande diferencial deste sistema é o fato dele ser genérico, tanto para melhor moldar-se às necessidades da organização que irá implantar, como também para o administrador do sistema fazer um melhor gerenciamento. Pela finalidade e pela genericidade do sistema, ele foi denominado de Help Desk Framework.

3.1 Organograma

A base inicial do trabalho foi a criação de um organograma. Foi necessário adequar a idéia de ser genérico e flexível, mas também de ser robusto para suportar uma estrutura muito complexa. Inicialmente, a modelagem do sistema baseou-se em objetos, pois simplifica o entendimento, por se aproximar da linguagem natural, além de ser o paradigma no qual está implementado, utilizando a linguagem de programação Java.

Iniciou-se a estrutura do organograma com o objeto Estrutura. Ele foi o ponto inicial de toda estrutura do organograma. Um objeto Estrutura contém um Nodo, que é o objeto que representa cada item do organograma. Um Nodo pode conter uma coleção de outros Nodos. Um Nodo também pode conter uma coleção de Elementos, que são as opções do Nodo. Na Figura 4 é ilustrado o diagrama de classes referente a Estrutura.

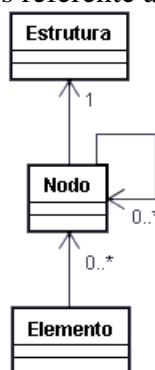


Figura 4. Diagrama de classes relativa a Estrutura.

Para um melhor entendimento sobre o objeto Estrutura, exemplificou-se na Figura 5 a estrutura de uma organização, tomou-se como exemplo o organograma da Escola Politécnica de Pernambuco (POLI). O objeto Estrutura foi todo o organograma da POLI. O Nodo principal que pertence ao objeto Estrutura é o Nodo denominado Diretoria. Ele contém alguns outros Nodos que são os Departamentos. Alguns dos Departamentos contém outros Nodos. O único departamento que não contém nenhum outro Nodo é o Departamento Básico. Então os outros departamentos contém outros Nodos que são os cursos.

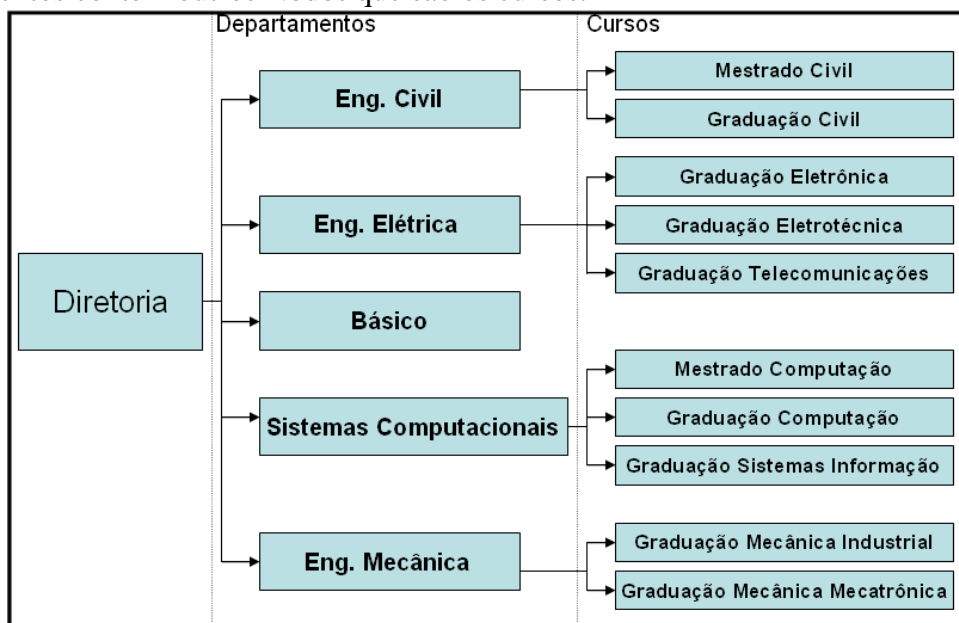


Figura 5. Exemplo do organograma da POLI.

Na Figura 6, expandiu-se o organograma da POLI somente para o Departamento de Sistemas Computacionais (DSC). Neste caso, a figura apresenta cinco objetos Elemento que são as opções do Departamento.

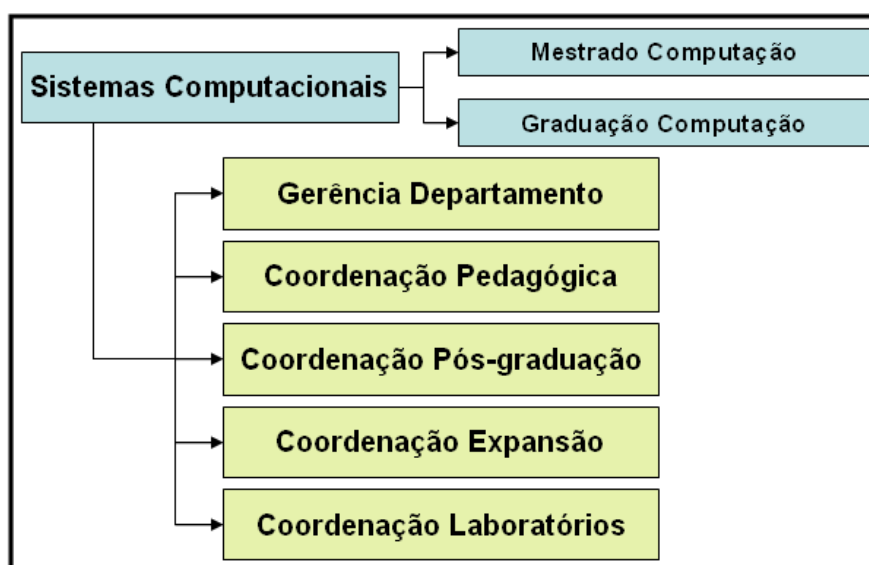


Figura 6. Exemplo do organograma do Departamento de Sistemas Computacionais.

3.2 Visão

A Visão é o modo que uma estrutura pode ser apresentada. Uma visão permite que a aplicação tenha alguns modos de exibição para entrada de chamados, com esses modos de exibição diferentes. Por exemplo, pode-se criar um modo de exibição para usuários externos ao sistema e outro para usuários internos. Para isso, são criadas Visões que são cada um desses modos de exibição. Uma mesma estrutura pode ter várias Visões diferentes, com finalidades diferentes, até podendo ser uma completamente diferente da outra.

Para a criação das Visões foi definido o objeto Visão. O objeto Visão vai conter uma referência ao objeto Estrutura que ele vai trabalhar e a um outro objeto, o ConfigHTML, que é o objeto responsável por inserir conteúdo HTML ao que vai ser exibido na saída da Visão. Na Figura 7 é exibido o diagrama de classes referentes à Visão. Ele mostra a relação do objeto Visão com os objetos Estrutura e ConfigHTML.

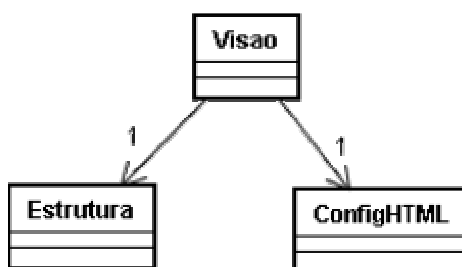


Figura 7. Diagrama de classes relativo a Visão.

O objeto Visão também contém uma lista de Nodos e Elementos disponíveis para a exibição na Visão. Então um Nodo pode estar disponível para uma Visão interna de uma organização, mas não estar disponível para uma Visão a clientes externos. É nessa lista que diz se o Nodo ou se o Elemento estar disponível para esta Visão.

Para exemplificar, é apresentado, na Figura 8, o organograma de uma empresa fictícia, a SuperVendas.com. É mostrado no Organograma duas Visões, a Visão Interna que simbolizada pela cor azul e Visão Externa representada pela cor amarela. A parte central esverdeada é a parte comum às duas Visões.

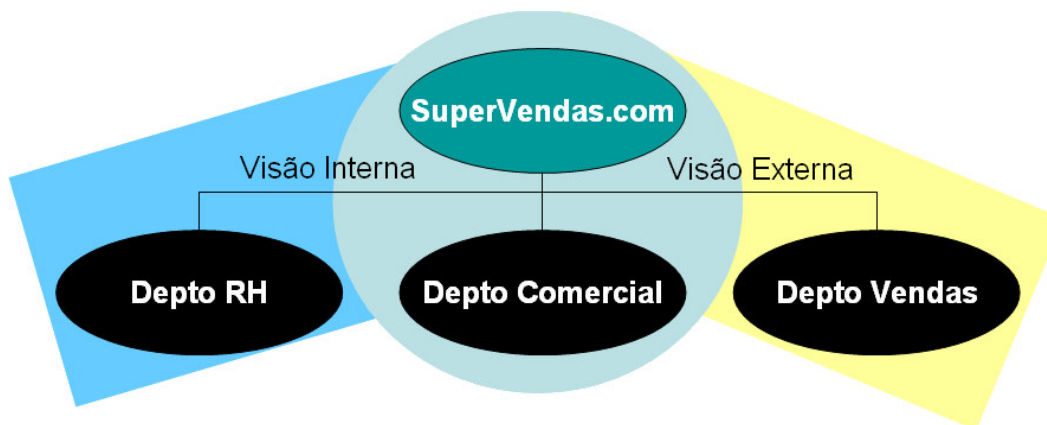


Figura 8. Organograma e duas Visões para uma mesma organização.

3.3 Formulário

O modo para dar entrada em novas Ocorrências é através dos formulários. Formulários são conteúdos HTML, seguindo padrões das páginas convencionais. O HTML vindo do formulário é inserido dentro da página quando o usuário desejar abrir uma nova Ocorrência, após selecionar um Elemento. Cada Elemento de uma Visão, deve possuir um formulário associado. O formulário contém os campos a serem preenchidos pelo usuário, Figura 9, e que serão inseridos na Ocorrência. Os formulários não estão mapeados como objetos, pois, com mencionado antes, eles estão da forma de HTML básico e apenas será inserido dentro da página.



Setor:
Comercial

Tipo:
Solicitação

Nome:

E-mail:

Assunto:

Escreva aqui:

Enviar Limpar

Figura 9. Exemplo de um Formulário.

Um objeto existente é o objeto `ElementoStatusFormulario`. É neste objeto que existe uma relação entre o Elemento que foi selecionado, para a abertura da Ocorrência, uma referência a um objeto Status (será visto no próximo tópico) e o nome do formulário a ser inserido.

Como exposto anteriormente, o Formulário é baseado em conteúdo HTML padrão, podendo ter vários campos. Isso deixa o sistema bastante genérico, uma vez que permite sua adaptação para diferentes necessidades. Outro fator importante é o de não amarrar o sistema a um padrão de entrada de dados. Por exemplo, em um formulário podemos ter alguns dados necessários como Nota Fiscal. Em outro formulário esse dado não é relevante, mas outros dados são mais importantes como o Lote Fabricação, por exemplo. Então poderemos ter formulários distintos, com dados distintos entre eles.

3.4 Estados

Para a tramitação de uma Ocorrência, sempre existem dois estados: sua abertura e seu fechamento. Essas duas possibilidades são os estados padrões da Ocorrência. Só que nem sempre uma Ocorrência vai necessitar somente dois estados. Podem existir casos onde haverá a necessidade de criar um ou mais estados intermediários. Para o projeto aqui descrito definiu-se o objeto Status, e ele é o objeto referente a todos os estados que uma Ocorrência pode ter.

Para dar mais flexibilidade ao sistema, foi definido no objeto Status duas possibilidades obrigatórias (Aberta e Fechada), podendo ter várias outras possibilidades a serem configuráveis. Além de o objeto Status ser configurável para ser aplicado a vários tipos de Ocorrências, também podem existir vários objetos Status distintos. Com isso aumenta a flexibilidade do sistema referente ao número de estados possíveis, não só definindo um tipo de caminho uma ocorrência pode seguir, mas vários caminhos possíveis.

Na Figura 10, exemplificaram-se dois Status diferentes que podem ser aplicados para uma mesma Estrutura. Na Figura 10 (a) mostra-se um objeto Status simples, contendo apenas três estados (Aberto, Tramitando e Fechado). Já na Figura 10 (b) contém seis estados (Aberto, Iniciado, Em Análise, Em Atendimento, Finalizando e Fechado). Nos dois casos, as máquinas de estado possuem sempre os estados obrigatórios que são Aberto e Fechado.

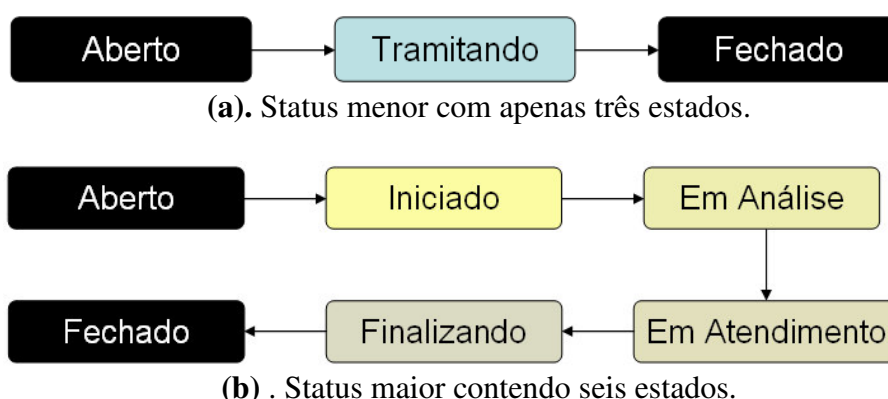


Figura 10. Exemplo de estados para dois Status.

3.5 Usuário

Quem vai interagir com o sistema vão ser pessoas. Na abordagem adotada pelo sistema, a modelagem orientada a objetos, foi definido o objeto Usuário como sendo a abstração para as pessoas que vão interagir com o sistema. Também foram definidos os perfis dos Usuários e suas possibilidades de interação. A seguir são apresentados os três perfis que foram definidos, e é feita um breve discussão sobre cada um.

- **Usuário Padrão:** Para esse perfil de usuário, ficou habilitado no sistema apenas a criação de novas Ocorrências e o acompanhamento das Ocorrências abertas por ele. Para o Usuário Padrão não será necessário autenticar direto no sistema, mais

em uma interface do sistema adicionado a algum Web Site da organização, pois o sistema prover uma interface externa, onde o usuário padrão interage fora do sistema principal, sem saber que é o Help Desk Framework o sistema que ele esta interagindo.

- **Usuário Especial:** O Usuário Especial tem as mesmas atribuições do usuário padrão, mais a possibilidade de responder, e dar andamento à Ocorrências no sistema. Para o andamento e continuidade das Ocorrências, é necessário o *Login* no sistema principal, pois é ele que provê essas funcionalidades.
- **Administrador:** Este perfil é para dar acesso a fazer todas as operações no sistema. O Administrador tem desde as atribuições do Usuário Padrão, quanto algumas atribuições específicas dele, como criar outros administradores. O Administrador pode modificar qualquer Ocorrência, fazendo as alterações necessárias além de cuidar da parte dos cadastros.

3.6 Ocorrência

Ocorrência é o principal foco deste sistema. A principal funcionalidade deste sistema é a abertura, andamento e resolução de uma Ocorrência. Pela genericidade do sistema, uma Ocorrência pode ser uma dúvida, uma reclamação, uma solicitação de compra, pedidos de reembolso, entre outros.

Para modelagem de objetos, como início tem o objeto principal, Ocorrência. O objeto Ocorrência é a referência no sistema para as Ocorrências. Quando um Usuário inicia uma nova Ocorrência na interface de abertura, vai ser adicionada no banco de dados uma referencia a um novo objeto Ocorrência. Um objeto Ocorrência tem os seguintes dados vinculados:

- **Usuário:** O usuário que entrou com os dados e criou a Ocorrência. Com o objeto Usuário, o sistema vai saber dos dados do Usuário para acesso ao andamento da Ocorrência e como entrar em contato o Usuário. Vai ser mostrado no próximo capítulo que pode vincular os dados dos Usuários de outros sistemas da organização, caso já contenha uma tabela de Usuários. Também é possível a criação de um novo cadastro de Usuários.
- **Data e Hora:** A data e a hora que a Ocorrência foi aberta.
- **Status:** A referência ao objeto Status que a Ocorrência ira seguir. É a maquina de estados da Ocorrência. Com essa referência ao objeto Status é possível saber o ciclo de vida que a Ocorrência pode ter.
- **Status Atual:** O índice indicando em qual estado do Status a Ocorrência se encontra no momento. Como é guardada uma referência ao objeto Status, precisa deste índice para saber em qual estado, dos estados existentes, o objeto Status estar referenciando no momento.
- **Formulário:** Esse campo indica qual o Formulário que a Ocorrência foi aberta.
- **Elemento:** O Elemento pelo qual a Ocorrência foi aberta. Como uma Ocorrência é aberta através da navegação pela Estrutura, também é de interesse saber a qual foi o Elemento da Estrutura que originou este campo. Essa informação é de interesse tanto para a tramitação da Ocorrência, quanto também para futuras informações estatísticas, relatórios, criação e exibição de FAQ.

- Campos: Esse item é bastante importante. Ele é uma lista de vários campos vindos do Formulário, com o nome do campo e os respectivos valores. Esses dados são de fundamental importância para o entendimento da Ocorrência, pois neles conterá todos os detalhes, textos inseridos e o que for primordial para o entendimento, importância, tramitação e relevância da Ocorrência.

Outro dado importante que vai ser persistido é o objeto OcorrenciaCaminho. Para cada passo do Status, Usuários e definições de uma Ocorrência, será gerado um objeto OcorrenciaCaminho, para manter todo o histórico da Ocorrência. Isso serve tanto para organização e gerenciamento interno de quem cuida das Ocorrências, quanto também para o Usuário que criou a Ocorrência saber por quem ou onde sua Ocorrência teve passagem.

Na Figura 11 é apresentado o diagrama de classes para o objeto Ocorrência e seus objetos relacionados.

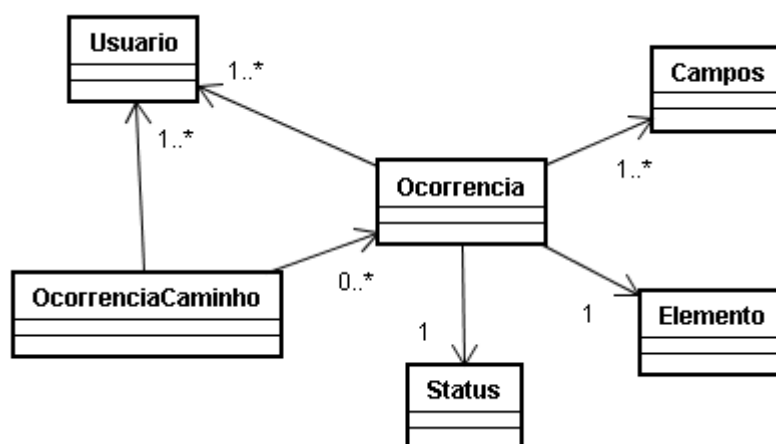


Figura 11. Diagrama de classe relativo a Ocorrência.

3.7 FAQ

O mecanismo de FAQ é uma listagem de soluções de Ocorrências já solucionadas ou previamente definidas a fim de tentar resolver a necessidade de uma Ocorrência, sem precisar a criação de uma nova. Essas Ocorrências da FAQ é uma consulta rápida e vai conter os motivos que levaram a intenção de abrir uma Ocorrência e suas soluções possíveis, definidas anteriormente ou através de experiências dentro do sistema.

Capítulo 4

Implementação

O conteúdo deste capítulo trata como a modelagem foi implementada. Para um melhor entendimento e organização, este capítulo ficou dividido em três partes. A primeira aborda a montagem da estrutura em XML [5], onde quem for implantar o sistema na organização também deve iniciar o trabalho. A segunda trata da implementação no lado da ferramenta que realiza os cadastros e acompanhamento das Ocorrências. E a terceira refere à como a implementação está disponibilizando a interface para ser utilizada no Web site da organização que irá implantar o sistema.

4.1 Estrutura

Toda a parte da Estrutura foi criada em objetos, sendo armazenado em arquivos XML. O intuito é facilitar o trabalho de modelagem e leitura do desenvolvedor e implantador, bem como salvar as configurações em arquivos facilmente editáveis, que são os arquivos XML.

O *parser* utilizado para leitura e escrita destes arquivos XML, foi o XStream [15]. Com este *parser* o desenvolvedor não precisa se preocupar com o conteúdo dos objetos e nem com o conteúdo do XML. Basta fazer o mapeamento de cada classe Java para uma *tag* XML correspondente. No restante do texto foi abstraída a questão relacionada com a criação dos arquivos XML, pois o foco será a leitura do XML e seus objetos criados a partir da leitura dos arquivos.

A partir de um arquivo “estrutura.xml” o *parser* vai fazer o mapeamento do conteúdo em um objeto Estrutura. Dentro deste objeto estrutura vai conter dois atributos: o primeiro é o nome da Estrutura e o segundo é seu identificador (ID), que tem que ser único para toda Estrutura. Deve existir dentro do Objeto Estrutura, uma referência a um objeto Nodo. Esse objeto Nodo também conterá um ID e o seu nome. A partir deste objeto Nodo, todos os objetos internos ao Nodo serão: uma lista de outros objetos Nodo e uma lista de objetos Elemento.

Na Figura 5 apresenta-se um exemplo de objetos Estrutura, Nodo, lista de objetos Nodo e lista de objetos Elemento. O objeto Estrutura é todo o organograma da POLI. Dentro do objeto Estrutura encontra-se a Diretoria como o seu Nodo interno. Os Departamentos são a lista de Nodos interna ao Nodo Diretoria. Na Figura 6 é mostrado mais detalhadamente o organograma do DSC. Neste caso é visualizado dois Nodos que são a área de graduação e a área de mestrado. Também na Figura 6 é visto uma lista de Elementos, que são a Gerência e as

Coordenações. Na Tabela 2 é exibida uma listagem com alguns desses objetos mapeados na Figura 5 e Figura 6.

Tabela 2. Mapeamento dos objetos e propriedades, da Estrutura, referentes a Figura 5 e Figura 6.

Nome	ID	Objeto	Objeto Interno à (ID)
POLI	1	Estrutura	Nenhum
Diretoria	2	Nodo	1
Engenharia Civil	3	Nodo	2
Básico	5	Nodo	2
Sistemas Computacionais	6	Nodo	2
Mestrado Computação	61	Nodo	6
Graduação Computação	62	Nodo	6
Gerência Departamento	601	Elemento	6
Coordenação Pedagógica	602	Elemento	6
Coordenação Pós-Graduação	603	Elemento	6
Coordenação Extensão	604	Elemento	6
Coordenação Laboratórios	605	Elemento	6

Após o objeto Estrutura ser lido em memória, agora é a vez do objeto Visão. Podem haver mais de um objeto Visão, disponíveis para ler uma mesma Estrutura. Um objeto Visão também é armazenado em XML. O conteúdo do objeto Visão inicia com uma referência ao objeto Estrutura. É daí que se vem o nome Visão, pois o objeto Visão nada mais é do que um modo de exibição para uma Estrutura. Dentro do objeto Visão existe uma lista de ID's que fazem referência a objetos Nodos e Elementos da Estrutura. Com isso, só será disponibilizado em uma Visão os Nodos e Elementos cujos ID's estiverem listados dentro da lista, no objeto Visão.

Outro aspecto do objeto Visão é que ele não precisa ser único como o objeto Estrutura. Uma mesma Estrutura pode ser exibida por objetos Visões diferentes. Um exemplo bom disto é mostrado na Figura 8. O que é necessário para isso é criar dois arquivos XML distintos, duas Visões, e cada uma delas ter suas próprias configurações.

Também dentro do objeto Visão existe outra lista com objetos ElementoStatusFormulário. Esse objeto é uma relação para três itens da configuração:

- Elemento: O elemento da Estrutura que foi selecionado para abertura de uma nova Ocorrência.
- Status: O nome do objeto Status que o Elemento vai utilizar como máquina de estados.
- Formulário: Qual formulário vai ser utilizado para abertura de uma nova Ocorrência.

Então, ao selecionar um Elemento de uma Estrutura, para abertura de uma nova Ocorrência, vai ser procurado nesta lista de ElementoStatusFormulário qual o objeto Status e qual será o Formulário utilizado para a abertura desta nova Ocorrência. Essa relação confere uma grande flexibilidade ao sistema, permitindo que um mesmo formulário seja aplicado a vários Elementos diferentes, podendo também ter várias máquinas de estado diferentes, a depender da necessidade e configuração.

O último elemento na implementação da Visão é o objeto ConfigHTML. Ele contém uma variedade de variáveis que serão utilizadas para inserir entre os conteúdos da Estrutura, durante uma navegação no Web Site que for implementar o sistema. Estas variáveis, listadas no Apêndice A, devem ser utilizadas desde a descrição do nome da Estrutura, Nodo e Elemento, até o nome da função JavaScript que será chamada ao clicar no item. As variáveis do objeto ConfigHTML também são aplicadas a FAQ.

Na Figura 12 é mostrado um exemplo da utilização destas variáveis. Na Figura 12 (a) mostra o conteúdo desejado com HTML. Na Figura 12 (b) é mostrado a variável e como conseguir pegar este conteúdo dentro do objeto ConfigHTML, e na Figura 12 (c) é mostrado o resultado desejado no navegador Web.

```
<tr><td><span class="style3" onClick="abreNodo(21);">
    Recursos Humanos
</span></td></tr>
<tr><td><span class="style3" onClick="abreNodo(22);">
    Comercial
</span></td></tr>
```

(a). Conteúdo HTML desejado.

Constante	Método de Acesso	Valor
corpoAntesNodo	getCorpoAntesNodo()	<tr><td>
corpoFuncaoJavaScript	getCorpoFuncaoJavaScript()	abreNodo
corpoFontClasse	getCorpoFontClasse()	style3
corpoDepoisNodo	getCorpoDepoisNodo()	</td></tr>

(b). Constantes e como obter o valor.

SuperVendas.com
Departamentos
Recursos Humanos
Comercial

(c). O resultado esperado, quando visualizado em um Navegador Web.

Figura 12. Exemplo do uso das constantes do objeto ConfigHTML.

4.2 Ferramenta de Acompanhamento

Nesta sessão será tratada a implementação da ferramenta para o acompanhamento de Ocorrências. Inicialmente foi utilizado o servidor de aplicação HTTP, Apache [13] e o Tomcat [1]. O Tomcat é utilizado para o processamento de código Java. Para o uso de Java, na ferramenta de acompanhamento de Ocorrências, estar sendo utilizado *Java Server Pages* (JSP) [3]. JSP são conteúdos HTML incluindo códigos Java no interior.

A primeira página da ferramenta de acompanhamento é a página de *Login*. Nela o usuário especial ou administrador deverão fazer as autenticações com e-mail e senha, conforme

mostra a Figura 13. Esta página é a única que é carregada completamente pelo navegador Web. Ela é dividida em 4 partes, mostradas na Figura 14. O cabeçalho e rodapé (primeira e última partes respectivamente) são estáticas. A parte central esquerda fica a área de Menu. É nesta área que o Menu é carregado após ter feito o *Login*. E na parte central direita fica a área de trabalho, onde as funcionalidades vão sendo exibidas e acessadas pelo usuário.



Figura 13. Página inicial da ferramenta de acompanhamento de Ocorrências.

Após digitar e-mail, senha e pressionar o botão Entrar, uma função JavaScript vai tratar de pegar os dados dos campos preenchidos e enviar o conteúdo via HTTP *POST*², para uma página que centraliza todas as requisições. Todas as ações do usuário serão submetidas para a página “corpoAction.jsp” que verifica a ação escolhida pelo usuário e faz seu tratamento. Neste caso, o *Login*, a página “corpoAction.jsp” é quem verifica se os dados dos usuários estão corretos. Caso esteja correto, carrega somente a área de menu com as opções disponíveis para o usuário que fez o *Login* e carrega na área de trabalho a visualização disponível para esse usuário.

As ações de acessar os dados, enviá-los para a página que cuida do processamento e atualizar somente a área desejada, está sendo feita com Ajax [2]. Para essa ação de *Login* são feitas duas atualizações. A primeira é a do corpo que é enviado com os dados e feito a autenticação no sistema. A segunda é a área de menu, onde após a autenticação e validação dos dados é atualizada com o conteúdo disponível para o usuário autenticado. Com isso é necessário baixar apenas o conteúdo necessário, e não toda a página novamente.

² POST é um modo de envio de dados de um formulário, onde os dados vão no corpo de um pacote HTTP, diferentemente do método GET que os dados vão na URL do endereço.

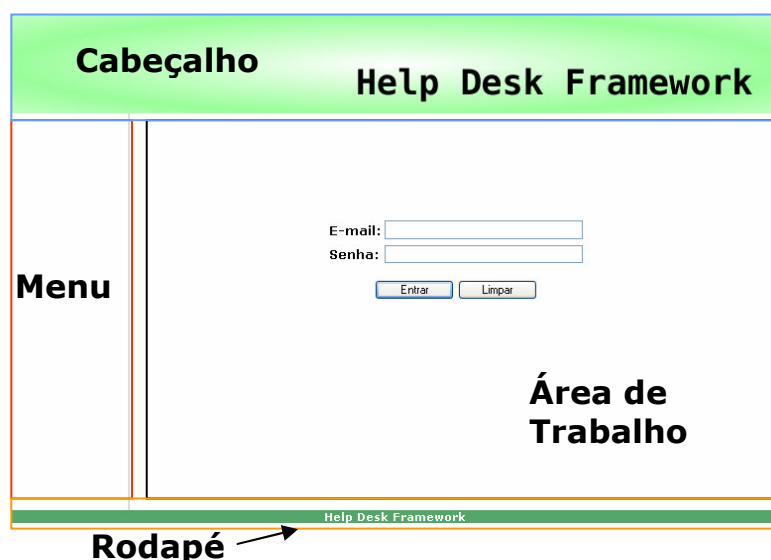


Figura 14. Divisões das partes da página inicial.

As opções mostradas no Menu Principal e disponíveis para o usuário estão listadas na Tabela 3. Algumas das opções estão disponíveis somente para o Administrador, outras também para o Usuário Especial, mas com algumas ressalvas. Vale lembrar que só podem fazer *Login* na ferramenta de acompanhamento os Administradores e o Usuário Especial.

Tabela 3. Funcionalidades do sistema e qual tipo de usuário pode utilizar.

Funcionalidade	Referente	Administrador	Usuário Especial
Listar	Ocorrência	Pode listar todas as Ocorrências e de todos os usuários.	
Editar		Pode editar todas as Ocorrências e de todos os usuários. Inclusive ocorrências já finalizadas.	Pode editar somente as suas Ocorrências. Não pode editar Ocorrências já finalizadas.
Criar Nova	FAQ	Pode criar editar qualquer FAQ para qualquer departamento	
Editar / Excluir			
Criar Novo	Usuário	Pode criar e editar qualquer usuário com qualquer perfil.	Pode criar e editar somente Usuários Especiais e Usuários Padrão.
Editar Excluir			
Usuário / Elemento	Usuário/ Elemento	Pode criar vínculos entre Administradores e Usuários Especiais com Elementos.	Pode criar vínculos entre Usuários Especiais com Elementos.

A funcionalidade Listar exibe uma lista com todas as Ocorrências que estão no tipo de listagem. Em Listar Novos, apresenta as Ocorrências que acabaram de ser criadas e ainda não ocorreu nenhuma modificação para ela. Como segunda opção, Listar Abertas mostra as que já passaram pelo menos por algum usuário, fazendo seu comentário, ou passando para outro usuário, mas ainda não foram finalizadas. A opção Listar Fechadas mostra as Ocorrências já finalizadas pelos Usuários responsáveis. Por fim a opção Listar Todos mostra as todas as ocorrências cadastradas pelo sistema.

Ao acessar uma listagem citada acima, o Administrador visualizara todas as Ocorrências que estão no Status solicitado. Já o Usuário Especial só terá acesso as Ocorrências que estiverem atribuídas a ele, exceto para a opção Listar Todas, que é apenas de visualização.

Após ser selecionada uma Ocorrência em uma listagem, no momento que for visualizada, aparecerá além dos campos inseridos pelo Usuário que abriu a Ocorrência, também todos os passos que a Ocorrência seguiu até o momento (com o Status, Usuário e comentário em cada momento) e opções para comentar, modificar status e direcionar para um próximo usuário. Na Figura 15 é mostrada a tela de edição de Ocorrências. Ela mostra os campos inseridos pelo usuário solicitante, o histórico e os dados para modificação da mesma. Na Figura 15 não estar sendo visualizada os comentários, pois para visualizar, pois tem que clicar em cima da linha e será aberta uma janela *popup* com a descrição.

Editar Ocorrência

Campos

Campo Nome	Campo Valor
nome	Jorge Silva
email	jorgesilva@hotmail.com
descricao	Queria Informações sobre assistência técnica de um Celular que foi comprado por mim em no site www.supervendas.com. Aparelho Nokia 6020. Nota Fiscal Número: 1234567
telefone	(81)99851234

Histórico

Usuário	Encaminhou Para	Status
CordenaçãO do Departamento	Julio Cesar Taveira	[Aberto] -> Tramitando
Julio Cesar Taveira	Julio Cesar Taveira	Tramitando -> Fechado

Dados para Modificação

Status:

Encaminhar Para:

Comentário:

Figura 15. Página para Edição de uma Ocorrência.

As telas com a listagem das Ocorrências, como sua edição, são sempre atualizadas na área de trabalho, utilizando funções JavaScript, e sendo enviadas pelo servidor e devolvidas para o navegador Web, através de Ajax. Outra aplicação das tecnologias descritas no Capítulo 2, é a persistência de dados.

As interações da aplicação com os dados salvos das Ocorrências são tratados diretamente com objetos. Existe aí o objeto Ocorrência com todos os seus dados. Para salvar e resgatar esses objetos do banco de dados MySQL [14] é utilizado o Hibernate [4]. O Hibernate faz um mapeamento dos dados para disponibilizar para a aplicação apenas trabalhar com os objetos, sem a necessidade de realizar consultas e atualizações da tabela, através de comandos SQL's.

Outra grande vantagem para aplicações utilizando Hibernate e que agrega para o Help Desk Framework, é que o desenvolvedor não precisa se preocupar com o banco de dados escolhido. No desenvolvimento do Help Desk Framework utilizou-se o MySQL, mas pode ser facilmente migrado para SQL Server, Oracle entre outros, que a aplicação não vai sofrer modificações. Basta mudar o arquivo de configuração do Hibernate (hibernate.cfg.xml) e copiar a biblioteca de funções para o banco de dados desejado. A aplicação também ficou bastante genérica pela versatilidade de utilizar algumas tabelas em banco de dados já existente na organização. Essa utilidade se dá pela possibilidade de reutilizar a tabela de cadastro de usuários, invés de criar um novo cadastro com os usuários que serão utilizados no sistema. Isto é feito modificando o arquivo Usuário.hbm.xml para selecionar a tabela já existente, bem como fazer o mapeamento dos campos referentes aos dados do usuário. A Figura 16 mostra as modificações necessárias a fim de mudar o banco de dados e a tabela de usuário. Na Figura 16 (a) são mostrados os dados da conexão com o banco de dados, qual *driver* está sendo utilizado, o usuário e a senha do banco de dados. Já na Figura 16 (b) é mostrado o nome da tabela mapeada para o objeto Usuário e quais campos estão sendo utilizados para trabalhar na aplicação Help Desk Framework.

```
<property name="connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="connection.url">jdbc:mysql://localhost:3306/hdfframework</property>
<property name="connection.username">root</property>
<property name="connection.password"></property>
```

(a). Modificações a serem feitas para modificar o SGBD utilizado na aplicação.

```
<class name="br.com.jtsistemas.hdf.persistencia.Usuario" table="hdf_usuario">
  <id name="id" column="id_usuario">
    <generator class="native"/>
  </id>
  <property name="nome"/>
  <property name="email"/>
  <property name="senha"/>
  <property name="endereco"/>
  <property name="cidade"/>
  <property name="tipo"/>
</class>
```

(b). Modificações a serem feitas para modificar a tabela de usuário.

Figura 16. Locais nos arquivos de configurações para modificação no banco de dados.

Para os cadastros de FAQ, Usuário e o vínculo Usuário/Elemento, também são utilizadas as mesmas abordagens citadas anteriormente. Estão sendo utilizadas funções JavaScript após a ação do usuário e o envio/recebimento de informações através de Ajax, bem como a utilização do mapeamento de objetos em Hibernate para fazer a persistência de dados.

No cadastro de FAQ, o usuário deve selecionar qual o elemento que quer inserir um registro de FAQ, colocar a pergunta e a resposta para a pergunta. Então, a partir deste momento, o Elemento da Estrutura vai conter mais uma FAQ para ele no cadastro.

Para os usuários, a adição no cadastro é feita primeiramente pela verificação da existência do e-mail no sistema. Caso contrário, o usuário deve digitar os dados no novo Usuário e salvá-lo. Para a edição e exclusão, o início também se dá com o fornecimento do e-mail. Após verificação da existência do e-mail, os dados do usuário são exibidos para alterar ou excluir o usuário.

Já o vínculo de Usuário/Elemento é utilizado para destinar novas Ocorrências. Então, quando uma nova Ocorrência é criada, uma pesquisa nos objetos UsuarioElemento é feita, a fim de saber qual é o usuário responsável pela nova Ocorrência, daquele Elemento. Durante o cadastro de vínculos Usuário/Elemento, primeiramente é solicitado o Usuário. Logo após são listados todos os vínculos já existentes para o Usuário e os Elementos possíveis para vínculos. Então o usuário que estiver modificando o cadastro, pode tanto inserir novos vínculos, como também remover vínculos já existentes.

4.3 Interface Externa

Esta terceira parte da implementação, diz respeito à interface que o sistema Help Desk Framework disponibiliza para ser implantada no Web site da organização que vai implantar o sistema. Para quando o usuário que deseja abrir uma nova Ocorrência, é mais familiar para ele trabalhar com a interface gráfica do Web site da organização, do que interagir diretamente com o sistema. É essa disponibilização de serviços a serem adicionados no Web site que é a interface externa do Help Desk Framework.

Essa interface são formadas por *servlets* [3] Java que recebem dados sobre a Estrutura, Visão, *Login*, cadastros de usuários e o mais importante, a criação de Ocorrências. *servlets* são classes Java que são mapeadas a endereços no arquivo web.xml, que fica na pasta WEB-INF do Apache Tomcat. Quando um caminho deste mapeamento é chamado, uma classe Java é chamada para fazer o tratamento da execução solicitada.

Na Tabela 4 estar listado os *servlets* implementados no sistema Help Desk Framework e suas funcionalidades são explicadas abaixo.

Tabela 4. *Servlets* disponibilizado na interface disponibilizado externamente pelo Help Desk Framework.

Classe Servlet	Caminho	Referente
EstruturaServlet	/servlet/estrutura	Estrutura e Visão
FormularioServlet	/servlet/formulario	Formulário
LoginExternoServlet	/servlet/loginExterno	Login
OcorrenciaServlet	/servlet/ocorrencia	Ocorrência
UsuarioServlet	/servlet/usuario	Usuário

O primeiro *servlet* mostrado na Tabela 4, o EstruturaServlet, permite ao usuário que deseja fazer a navegação pelo navegador Web e criar uma Ocorrência, conseguir chegar ao Elemento desejado para essa criação. Esse Servlet prove três serviços diferentes: estrutura, formulário e FAQ. Esses serviços devem ser passados como parâmetros no momento da chamada ao *servlet*. A seguir são descritos os três serviços:

- O primeiro serviço, o estrutura é responsável pela navegação. Ao ser solicitado o usuário deve passar o ID da Estrutura, do Elemento ou do Nodo. Ele vai retornar duas partes. A primeira parte com todo caminho percorrido até o elemento

selecionado. A segunda parte as opções do momento. Essas opções são Nodos ou Elementos. As duas partes estão separadas por uma variável do objeto ConfigHTML, que esta listada no Apêndice A.

- Já o segundo serviço é mostrar o Formulário de um elemento. Após chamar esse *servlet*, passando o ID do Formulário, é aberto, onde o implantador escolheu, o conteúdo que o Servlet retornou. O nome do Formulário é retirado do argumento de uma função definida em uma variável no objeto ConfigHTML.
- O terceiro serviço é o de FAQ. Fazendo uma requisição ao *servlet*, passando com parâmetro o ID do Elemento, vai ser retornado o conteúdo das FAQ's para o Elemento.

Na Figura 17 são mostrados alguns exemplos do retorno dos serviços que o *servlet* EstruturaServlet disponibiliza. Na Figura 17 (a) é apresentado um exemplo com o caminho percorrido e as opções do DSC. Já na Figura 17 (b) é mostrado um formulário de abertura de Ocorrência. E na Figura 17 (c) mostra o resultado do serviço FAQ.

POLO -> Diretoria -> Departamento de Sistemas Computacionais <ul style="list-style-type: none"> • >> Gerência • >> Coordenação Pedagógica Departamento • >> Coordenação Pedagógica Pós • >> Coordenação Extensão • >> Coordenação Laboratórios

(a). Exemplo do serviço estrutura do *Servlet* EstruturaServlet.

Formulário de Informações - DSC

Aluno:

Matricula:

Dúvida:

(b). Exemplo do serviço formulário do *Servlet* EstruturaServlet.

<ol style="list-style-type: none"> 1. Como posso ser aluno do DSC? <ul style="list-style-type: none"> • Para ser aluno do DSC tem que prestar vestibular para Engenharia da computação ou Sistemas de Informações. 2. O que é preciso para transferência externa? <ul style="list-style-type: none"> • É preciso ter acesso ao manual publicado periodicamente pela EPP. 3. O que é um aluno bloqueado? <ul style="list-style-type: none"> • Quando o aluno reprova ou não cumpre alguma disciplina de algum período pelo qual passou.

(c). Exemplo do serviço FAQ do *Servlet* EstruturaServlet.

Figura 17. Página para Edição de uma Ocorrência.

Para o segundo *servlet* (FormularioServlet) mostrado na Tabela 4, há uma funcionalidade menor, mas não menos importante. Ele é responsável por todos os dados vindos do Formulário e criar a nova Ocorrência. Dependendo se conseguiu criar a nova Ocorrência ou não, é mostrada uma página que deve estar configurada em uma constante do arquivo `application.properties`. Esse arquivo `application.properties` tem algumas constantes responsáveis por caminhos de arquivos e outras constantes, mostradas no Apêndice B.

O terceiro *servlet*, LoginExternoServlet, mostrado na mesma Tabela 4 é utilizado para entrar com os dados de e-mail e senha, a fim de autenticar o usuário para abrir a Ocorrência. Se os dados do *servlet* estiverem corretos, será criada na sessão uma variável para ser incluída no momento que o usuário estiver salvando a nova Ocorrência. Essa variável de sessão é diferente da variável utilizada no login para a ferramenta de acompanhamento. Ela somente dá acesso à criação de nova Ocorrência.

Outro *servlet* bastante importante é o OcorrenciaServlet. Ele permite ao Usuário que criou a Ocorrência acessar os dados de tramitação e desfecho. Ele disponibiliza três serviços, listados abaixo:

- Listar: é o serviço que faz a listagem de todas as ocorrências do usuário. Não precisa ser passado nenhum parâmetro para esse serviço. Ele retorna uma lista com todas as Ocorrências, utilizando a formatação de variáveis do objeto `ConfigHTML`.
- Mostrar: esse tem a responsabilidade de mostrar os dados para uma Ocorrência. O código da Ocorrência deve ser informado como parâmetro, no momento da solicitação do serviço deste *Servlet*.
- Descrição: para esse serviço o retorno é mais simples. Ele é responsável apenas pelo retorno ao comentário feito durante uma etapa da tramitação de uma Ocorrência.

Para o último *servlet*, UsuarioServlet, ficou o serviço de criação de um usuário, caso não exista, para poder criar novas Ocorrências. Ele recebe como parâmetro e-mail, nome, senha, endereço e telefone, e retorna se o usuário foi bem criado ou não.

Capítulo 5

Conclusões e Trabalhos Futuros

O intuito desde trabalho foi desenvolver uma ferramenta de abertura e acompanhamento de Ocorrências. Foi proposto também que o sistema fosse genérico e robusto, para uma melhor adaptação ao modo da organização que desejar implantar.

5.1 Conclusões

O resultado deste trabalho de conclusão foi uma ferramenta de acompanhamento e abertura de ocorrência. A mesma é bastante genérica, tanto para a adaptação a estrutura organizacional de quem a for implantar, quanto pela possibilidade da mesma moldar-se à infraestrutura existente. Além disso, a ferramenta permite o reuso do cadastro de funcionários e a não dependência do banco de dados adotado no desenvolvimento da aplicação.

A ferramenta também tem uma flexibilidade no cadastro do organograma da organização, na criação dos formulários de abertura de Ocorrências, na máquina de estados das Ocorrências e na disponibilização de interagir com a interface existente no website da organização.

Um fator interessante da tecnologia adotada foi o custo. Todas as aplicações e ferramentas de desenvolvimento têm custo zero, ou um valor financeiro muito baixo. O que é mais um atrativo para a implantação da ferramenta.

5.2 Trabalhos futuros

Novas funcionalidades podem ser implantadas, desde a estrutura da aplicação até a interface gráfica. A idéia é buscar um aperfeiçoamento contínuo.

Outras importantes modificações podem vir a partir do momento da implantação em alguns clientes e do *feedback* fornecido pelos mesmos.

Também é sempre bom estar atualizando a ferramenta em termos tecnológicos, a fim de sempre está ganhando robustez inovações. É importante considerar o custo para essas atualizações tecnológicas, a fim da aplicação não perder o seu atrativo pelo baixo custo.

Uma ferramenta que agregaria muito a esta aplicação é uma ferramenta específica para edição de conteúdo XML. Hoje a estruturação é baseada em conteúdo XML e editado diretamente no código. Uma interface gráfica amigável seria bastante útil para estruturar o organograma, as Visões, as variáveis da classe ConfigHTML, entre outros elementos organizado nos arquivos XML.

Outra importante evolução da ferramenta é a sua internacionalização. Já que é utilizando Java, seria bastante interessante a exploração do ResourceBundle³ para tornar a aplicação uma plataforma multi idiomas.

Uma funcionalidade que somaria, junto a FAQ, seria uma pesquisa de nos dados das Ocorrências já existentes. Seria bom adotar um mecanismo que faz parte de uma técnica de Inteligência Artificial (IA) chamada Raciocínio Baseados em Casos (RBC) [16]. Essa técnica baseada em casos é usada para encontrar soluções para um novo problema usando soluções adotadas em problemas similares já resolvidos.

³ ResourceBundle é uma classe que carrega arquivos de propriedades em Java, podendo utilizar vários idiomas a depender da necessidade e implementação.

Bibliografia

- [1] Apache Tomcat: The Apache Software foundation. Disponível em: <http://tomcat.apache.org/tomcat-5.5-doc/index.html>. Acessado em 23/05/2007.
- [2] ASLESON, R. and ASLESON, N. **Foundations of Ajax**, EUA: Apress, 2006, 266 p.
- [3] BASHAN, Brian. Use a Cabeça!: JSP & Servlets, São Paulo: Alta Books, 2005, 534p.
- [4] BAUER, Christian e KING, Gavin. Hibernate em Ação. São Paulo: Ciência Moderna, 2005, 530p.
- [5] BIRBECK, M. Professional XML. São Paulo: Ciência Moderna. 2001, 1266p.
- [6] COELHO, Á. V. S. ; FERNEDA, E. . Tecnologias para o Desenvolvimento de Sistemas Help Desk. In: VIII Semana de Informática da UFBA, 2000, Salvador. Anais da VIII Semana de Informática da UFBA, 2000. v. Único. p. 319-333.
- [7] DEITEL, H. e DEITEL, Paul. Java Como Programar. 4ª ed. São Paulo: Bookman, 2004, 1386 p.
- [8] DEITEL, H. e DEITEL, Paul. Internet Como Programar. 2ª ed. São Paulo: Bookman, 2004, 1274 p.
- [9] GOODMAN, Danny. JavaScript: A bíblia: São Paulo: Campus, 2001, 994p.
- [10] HOUAISS, A. Dicionário Houaiss da Língua Portuguesa. São Paulo: Objetiva, 2001. 3008 p.
- [11] NEtCraft: About NetCraft. Disponível em: <http://news.netcraft.com/about-netcraft>. Acessado em 23/05/2007.
- [12] NetCraft: Disponível em http://news.netcraft.com/archives/web_server_survey.html; Acessado em 23/05/2007.
- [13] RICH B. The Definitive Guide to Apache Mod_rewrite. United States: Apress, 2002, 142p.
- [14] SUEHRING, S. MySQL: a Bíblia. São Paulo: Campus, 2002, 732p.
- [15] XStream: Xstream Project. Disponível em <http://xstream.codehaus.org/>. Acessado em: 23/05/2007.
- [16] WANGENHEIM, Christiane G.; WANGENHEIM, Aldo. Raciocínio Baseado em Casos. Curitiba: Editora Manole, 2003. v. 1. 296 p.

Apêndice A

Tabela de Variáveis da classe ConfigHTML

Variável	Aplicação
inicioTudo	Conteúdo HTML a ser inserido no início da estrutura.
fimTudo	Conteúdo HTML a ser inserido no fim da estrutura.
separaCorpoTopo	Conteúdo HTML a ser inserido para separar o topo e o corpo na solicitação da estrutura.
antesXMLTopo	Conteúdo HTML a ser inserido no início do topo.
depoisXMLTopo	Conteúdo HTML a ser inserido no final do topo.
topoFontClasse	Estilo CSS utilizado para cada item do topo.
topoFuncaoJavaScript	Função JavaScript chamada ao clicar em um item no topo.
topoDivisorItens	Conteúdo HTML a ser inserido entre dois itens do topo
antesXMLCorpo	Conteúdo HTML a ser inserido no início das opções da estrutura..
depoisXMLCorpo	Conteúdo HTML a ser inserido no final das opções da estrutura.
corpoFontClasse	Estilo CSS utilizado para cada item das opções da estrutura.
corpoFuncaoJavaScript	Função JavaScript chamada ao clicar em um item das opções da estrutura.
corpoAntesNodo	Conteúdo HTML a ser inserido antes da exibição de um Nodo.
corpoDepoisNodo	Conteúdo HTML a ser inserido depois da exibição de um Nodo.
divisorNodoElemento	Conteúdo HTML a ser inserido entre os Nodos e os Elementos.
elementoClasse	Estilo CSS utilizado para cada item Elemento.
elementoFuncaoJavascript	Função JavaScript chamada ao clicar em um item Elemento.
corpoAntesElemento	Conteúdo HTML a ser inserido antes da exibição de um Elemento.
corpoDepoisElemento	Conteúdo HTML a ser inserido depois da exibição de um Elemento.
corpoAntesElementoList	Conteúdo HTML a ser inserido antes da exibição de uma lista de Elementos.
corpoDepoisElementoList	Conteúdo HTML a ser inserido depois da exibição de uma lista de Elementos.
antesFAQ	Conteúdo HTML a ser inserido antes da FAQ.
depoisFAQ	Conteúdo HTML a ser inserido depois da FAQ.
antesItem	Conteúdo HTML a ser inserido antes de um item da FAQ.
depoisItem	Conteúdo HTML a ser inserido depois de um item da FAQ.
antesPergunta	Conteúdo HTML a ser inserido antes de uma pergunta da FAQ.
depoisPergunta	Conteúdo HTML a ser inserido depois de uma pergunta da FAQ.
antesResposta	Conteúdo HTML a ser inserido antes de uma resposta da FAQ.
depoisResposta	Conteúdo HTML a ser inserido depois de uma resposta da FAQ.
divisorItem	Conteúdo HTML a ser inserido dividindo itens da FAQ.
faqVazia	Conteúdo HTML a ser inserido quando uma FAQ estiver vazia.

Apêndice B

Tabela de constantes do arquivo application.properties

Constante	Utilização
PASTA_TRABALHO	Pasta base, onde esta os arquivos .jsp e a pasta WEB-INF.
FORMS_PASTA	Pasta onde esta os formulários. Basta apenas colocar o caminho relativo a PASTA_TRABALHO.
XML_CONFIG_PASTA	Pasta onde esta os arquivos XML da Estrutura e Visão. Basta apenas colocar o caminho relativo a PASTA_TRABALHO.
CAMINHO	Caminho para a aplicação.
RESPOSTA_ABERTURA_OK	Arquivo HTML que deve ser mostrado, caso a abertura de um chamado foi bem sucedida.
RESPOSTA_ABERTURA_ERROR	Arquivo HTML que deve ser mostrado, caso a abertura de um chamado contém algum erro.
LOGIN_OK	Arquivo HTML a ser exibido, caso o login externo tenha sido bem sucedido.
LOGIN_ERROR	Arquivo HTML a ser exibido, caso o login externo tenha ocorrido algum erro.
LOGIN_EMAIL_INVALIDO	Arquivo HTML a ser exibido, caso o login externo tenha ocorrido algum erro.
LOGIN_SENHA_INVALIDA	Arquivo HTML a ser exibido, caso o e-mail esteja errado.
USUARIO_SESSAO_INVALIDA	Arquivo HTML a ser exibido, caso a senha esteja errada.
LISTA_PROCESSOS_NOVOS	Constante utilizada para indicar opção de Listar Ocorrência Novas.
LISTA_PROCESSOS_ABERTOS	Constante utilizada para indicar opção de Listar Ocorrências Abertas.
LISTA_PROCESSOS_FECHADOS	Constante utilizada para indicar opção de Listar Ocorrências Fechadas.
LISTA_PROCESSOS_GERAL	Constante utilizada para indicar opção de Listar Todas Ocorrências.

Apêndice C

Documento de Especificação de Casos de Uso

Help Desk Framework

Especificação de Casos de Uso

Responsável: Júlio Taveira

Conteúdo

1. Introdução	42
1.1 Propósito	42
1.2 Público Alvo	42
1.3 Convenções, termos e abreviações	42
2. Atores	42
3. Casos de Uso	43
3.1 Casos de Uso gerais	43
[UC001] Entrar no Sistema	43
[UC002] Cadastro de Usuário	43
3.2 Casos de Uso de configuração	44
[UC003] Configuração do Cadastro do Usuário	44
[UC004] Configuração da árvore estrutural	44
[UC005] Configuração dos formulários	45
[UC006] Vinculação de Formulários	45
[UC007] Configuração de status	46
3.3 Casos de Uso de execução	46
[UC008] Abrir uma Ocorrência	46
[UC009] Tramitar uma Ocorrência	47
[UC010] Listar Ocorrências	47
[UC010] Cadastro FAQ	48

Introdução

Propósito

Este documento especifica os casos de uso do sistema *Help Desk Framework* descrevendo os fluxos de eventos, entradas e saídas de cada caso de uso a ser implementado.

Público Alvo

Esse documento se destina para ser um guia no desenvolvimento do projeto, tanto para o desenvolvedor e ao orientador. Também tem finalidade de melhor entendimento sobre o projeto.

Convenções, termos e abreviações

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na tabela a seguir, estando apresentados por ordem alfabética.

Termo	Descrição
Atores	Alguém ou alguma coisa (ex: máquina, outro sistema), externa à aplicação, que interage de alguma forma com o sistema.
Casos de Uso	Uma seqüência de ações realizadas pelo sistema que resulta num valor observável para um ator específico.
Elementos	São itens de um nível organizacional. Em uma empresa pode ser expresso como um produto ou mesmo um serviço.
Nodos	Elemento referente à estrutura da árvore a nível organizacional. Na árvore estrutural pode ser uma folha, ou pode também ter filhos. Um nodo pode não ter ou ter vários Elementos.
Ocorrência	É a base deste trabalho. Ocorrência é a solicitação feita por um usuário, para obter informações ou resultado de algo desejado.
Requisitos funcionais	Requisitos técnicos do software que compõe o sistema, que descrevem ações que o sistema deve estar apto a executar, ou seja, o que o sistema deve fazer.
Requisitos não funcionais	Requisitos técnicos do software que compõe o sistema, que descrevem atributos que o sistema deve possuir ou restrições sob as quais ele deve operar.
Visão	Elemento que provê a saída de informações da Estrutura, para uma interface externa ao sistema.

Atores

Ator	Descrição
Administrador	Usuário que ira configurar e administrar o sistema. O Administrador tem acesso liberado para todas as áreas do sistema.

Usuário Padrão	Qualquer pessoa que utiliza o sistema. O Usuário Padrão só tem acesso à criação de Ocorrência, e acompanhamento das mesmas.
Usuário Especial	Usuário que irá tramitar com as Ocorrências, cuidar dos cadastros de usuários e FAQ.

Casos de Uso

Casos de Uso gerais

[UC001] Entrar no Sistema

Pré-condições: O web browser esta em uma pagina que não seja do sistema.

Pós-condições: O usuário esta verificando seu menu principal, dentro do sistema.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. O usuário digita a URL onde está instalado o sistema.
2. Uma pagina de autenticar o acesso é exibida para o usuário.
3. O usuário entra com o seu nome de acesso e sua senha.
4. O sistema faz a autenticação e mostra o menu principal disponível para o usuário.

Prioridade: Essencial Importante Desejável

[UC002] Cadastro de Usuário

Pré-condições: O usuário ter clicado em um link para ter acesso ao cadastro de usuário.

Pós-condições: Um novo usuário esta cadastrado.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Um formulário com os campos padrões de cadastro de usuário é mostrado.
2. O cadastrante coloca os dados no formulário.
3. Após feito as modificações desejada, o cadastrante salva as alterações.
4. A tela de confirmação é exibida.

Prioridade: Essencial Importante Desejável

Casos de Uso de configuração

[UC003] Configuração do Cadastro do Usuário

Pré-condições: O Administrador está autenticado no sistema e ter permissão de configuração.

Pós-condições: O formulário de cadastro de usuário esta configurado como desejado.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Administrador inicia a execução da configuração do cadastro de usuário.
2. Uma tela é mostrada ao administrador para escolher qual do tipo de usuário deseja configurar o formulário. Existe o usuário público e os usuários internos.
3. Depois de selecionado o tipo de usuário, o sistema mostra os campos padrões que podem ser marcados como:
 - a) Obrigatório: O cadastrante tem que adicionar o campo.
 - b) Opcional: Esse campo fica a vontade do cadastrante.
 - c) Não necessário: Esse campo não vai ser exibido para o cadastrante.
4. O Administrador pode colocar campos diferentes dos campos padrões.
5. Depois de feito as modificações desejadas, o usuário salva as alterações.
6. A tela de menu principal é exibida para o usuário e as modificações foram bem executadas.

Prioridade: Essencial Importante Desejável

[UC004] Configuração da árvore estrutural

Pré-condições: O Administrador está autenticado no sistema e ter permissão de configuração.

Pós-condições: A árvore estrutural estará de acordo com as necessidades para melhor utilização do sistema.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Usuário Configurador inicia a execução da configuração da árvore estrutural.
2. Uma tela com a árvore estrutural atual é mostrada.
3. O usuário pode inserir, alterar ou excluir novos Nodos e Elementos em qualquer parte hierárquica da árvore.
4. Após feito as modificações desejada, o usuário salva as alterações.
5. A tela de menu principal é exibida para o usuário e as modificações foram bem executadas.

Prioridade: Essencial Importante Desejável

[UC005] Configuração dos formulários

Pré-condições: O Administrador está autenticado no sistema e ter permissão de configuração.

Pós-condições: Os formulários estão criados corretamente e prontos para serem utilizados dentro do sistema.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Usuário Configurador inicia a execução da configuração de formulários.
2. Uma tela com os formulários atuais é mostrada.
3. O usuário pode inserir, alterar ou excluir os Formulários.
4. Após feito as modificações desejada, o usuário salva as alterações.
5. A tela de menu principal é exibida para o usuário e as modificações foram bem executadas.

Prioridade: Essencial Importante Desejável

[UC006] Vinculação de Formulários

Pré-condições: O Administrador está autenticado no sistema e ter permissão de configuração. Também existir uma árvore estrutural feita, formulários prontos, usuários internos cadastrados e status também cadastrado.

Pós-condições: Os formulários, usuários internos e status estão vinculados com Nodos e Elementos da árvore estrutural.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Administrador inicia a execução da vinculação de formulários.
2. Uma tela com a estrutura é mostrada.
3. O Administrador seleciona um nodo ou um elemento.
4. É mostrado pelo sistema o formulário e status já vinculado para aquele Nodo ou Elemento. O Administrador pode incluir ou remover algum formulário e status já existente.
5. Também é mostrado o(s) usuário(s) interno responsável(is) por aquele formulário e vinculo com a estrutura. O Administrador pode incluir ou remover algum usuário ao vinculo já existente. Além de poder ter vários formulários, também pode ter vários vínculos e usuários internos associados.
6. Após feito as modificações desejada, o Administrador salva as alterações.

7. A tela de menu principal é exibida para o usuário e as modificações foram bem executadas.

Prioridade: Essencial Importante Desejável

[UC007] Configuração de status

Pré-condições: O Administrador está esta autenticado no sistema e ter permissão de configuração.

Pós-condições: O status dos formulários está criado corretamente e prontos para serem utilizados dentro do sistema.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Administrador inicia a execução da configuração de status.
2. Uma tela com os status atuais é mostrada.
3. O usuário pode inserir, alterar ou excluir os status.
4. Após feito as modificações desejada, o usuário salva as alterações.
5. A tela de menu principal é exibida para o administrador e as modificações foram bem executadas.

Prioridade: Essencial Importante Desejável

Casos de Uso de execução

[UC008] Abrir uma Ocorrência

Pré-condições: Um usuário tem seu cadastro e estar autenticado no sistema. Uma Visão está disponível para o Usuário utilizar. Existe um formulário vinculado a um determinado Elemento e a um Status.

Pós-condições: Uma nova Ocorrência é adicionado no sistema e esta encaminhado para os responsáveis pelo mesmo.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. O usuário navega através da estrutura para encontrar o local para adicionar uma Ocorrência.
2. Pelo usuário é selecionado um elemento ou um nodo disponível para ser adicionada alguma Ocorrência.
3. O usuário digita os dados mostrados nos campos de uma nova Ocorrência.
4. O usuário salva as alterações da Ocorrência.

Prioridade: Essencial Importante Desejável

[UC009] Tramitar uma Ocorrência

Pré-condições: Uma Ocorrência esta disponível em Elemento e existe um ou vários especial ou administradores disponíveis para responder uma Ocorrência. Ou existe uma Ocorrência disponível diretamente para um usuário.

Pós-condições: A Ocorrência é respondida e o usuário padrão que deu entrada recebe um e-mail com o status da Ocorrência.

Casos de Uso relacionados: UC008.

Fluxo de eventos principal

1. Um usuário especial ou administrador responsável por um nodo ou elemento, seleciona uma Ocorrência disponível em um nodo ou elemento que ele é responsável, ou seleciona alguma Ocorrência ligada diretamente a ele.
2. Os dados da Ocorrência são exibidos.
3. O usuário especial ou administrador sua relevância sobre a Ocorrência.
4. Depois de ter feito os devidos comentários, o usuário finaliza a Ocorrência ou passa para outro nodo ou elemento competente, podendo passar para um determinado usuário
5. O usuário especial ou administrador salva as alterações da Ocorrência.

Prioridade: Essencial Importante Desejável

[UC010] Listar Ocorrências

Pré-condições: Várias Ocorrências estão cadastrada no sistema. O usuário especial ou administrador estar autenticado.

Pós-condições: As Ocorrências são mostrada, conforme solicitadas.

Casos de Uso relacionados: UC008.

Fluxo de eventos principal

1. Um usuário especial ou administrador seleciona uma das listagens disponíveis.
2. Uma lista de Ocorrência é exibida, a partir do tipo de exibição que o usuário especial ou administrador solicitou.
3. Os dados da Ocorrência são exibidos.
4. O usuário responde sua relevância sobre a Ocorrência.
5. O usuário salva as alterações da Ocorrência.

Prioridade: Essencial Importante Desejável

[UC010] Cadastro FAQ

Pré-condições: Alterações nas FAQ's de um determinado Elemento precisa ser feitas.

Pós-condições: As alterações foram bem sucedidas.

Casos de Uso relacionados: Nenhum.

Fluxo de eventos principal

1. Um usuário especial ou administrador seleciona uma das opções no cadastro de FAQ.
2. Os dados pertinentes a alteração e exclusão são exibidos..
3. O usuário especial ou administrador mas as modificações e inserções devidas.
4. O usuário salva as alterações da FAQ.

Prioridade: Essencial Importante Desejável

Apêndice D

Diagrama de Classes

