

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

SISTEMA DE GERENCIAMENTO DE CONTEÚDO DE
PÁGINAS WEB UTILIZANDO FLEX

MICHEL SCOZ

BLUMENAU
2007

2007/1-32

MICHEL SCOZ

**SISTEMA DE GERENCIAMENTO DE CONTEÚDO DE
PÁGINAS WEB UTILIZANDO FLEX**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciências
da Computação — Bacharelado.

Prof. Francisco Adell Péricas, Mestre - Orientador

**BLUMENAU
2007**

2007/1-32

SISTEMA DE GERENCIAMENTO DE CONTEÚDO DE PÁGINAS WEB UTILIZANDO FLEX

Por

MICHEL SCOZ

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Francisco Adell Péricas, Mestre – Orientador, FURB

Membro: _____
Prof. Oscar Dalfovo, Doutor – FURB

Membro: _____
Prof. Fabiane Barreto Vavassori Benitti, Doutora – FURB

Blumenau, 09 de julho de 2007

Dedico este trabalho aos meus pais, amigos e a todos que participaram diretamente ou indiretamente na realização deste.

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça.

Aos meus pais Osmar e Iraci pelo apoio durante toda minha vivência como acadêmico.

Aos meus amigos, pelos empurrões e cobranças.

À minha namorada Patrícia, pela paciência, compreensão e apoio durante todo desenvolvimento deste trabalho.

Ao meu orientador, Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho.

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação *web* para o gerenciamento de conteúdo de páginas na internet que visa facilitar e auxiliar a manutenção destes conteúdos, reduzindo o tempo gasto nestas atualizações e possibilitando que usuários leigos e sem conhecimentos de programação possam utilizá-lo sem maiores dificuldades. A especificação e implementação do trabalho utiliza FLEX, ActionScript 3, PHP5, MySQL e o *checklist* ErgoList para melhorias na intuitividade e facilidade de uso do aplicativo. Com o FLEX será possível desenvolver interfaces mais amigáveis e intuitivas para o gerenciador de conteúdo, e proporcionar maior desempenho na execução do aplicativo.

Palavras-chave: CMS. Web. FLEX. ActionScript. Gerenciamento. Conteúdo. Internet.

ABSTRACT

This work presents the development of a web application for content management of internet pages which aims to facilitate and help the maintenance of such contents, reducing the time spent on updating and it makes possible that users without programming knowledge can use it without major problems. The specification and implementation of the work uses FLEX, ActionScript 3, PHP5, MySQL and the checklist ErgoList to improve the intuitivity and facilitate the use of the application. With FLEX it'll be possible to develop friendlier and more intuitive interfaces for the content management system, and provide better performance when executing the application.

Key-words: CMS. Web. FLEX. ActionScript. Management. Content. Internet.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Quadro 1 – Exemplo da sintaxe de um arquivo MXML | 20 |
| Figura 1 – Exemplo da tela gerada pelo exemplo de código fonte do Quadro 1..... | 21 |
| Figura 2 – Fluxo de interações entre aplicação <i>web</i> e servidor | 22 |
| Quadro 2 – Exemplo de um trecho de definição de uma classe em ActionScript 3..... | 24 |
| Figura 3 – Diagrama de caso de uso de controle de acesso..... | 31 |
| Quadro 3 – Caso de uso UC01.01 Efetua Login | 33 |
| Figura 4 – Diagrama de casos de usos do sistema..... | 33 |
| Quadro 4 – Caso de uso UC02.01 Configurar Sistema | 34 |
| Quadro 5 – Caso de uso UC02.02 Adicionar Menu | 36 |
| Quadro 6 – Caso de uso UC02.03 Adicionar Conteúdo..... | 38 |
| Quadro 7 – Caso de uso UC02.04 Adicionar Galeria..... | 40 |
| Figura 5 – Diagrama de casos de uso do <i>website</i> | 40 |
| Quadro 8 – Caso de uso UC03.01 Visualiza Site | 41 |
| Figura 6 – Diagrama de atividades do sistema | 42 |
| Figura 7 – Diagrama de entidade e relacionamento | 43 |
| Figura 8 – Diagrama de classe da interface inicial..... | 44 |
| Figura 9 – Diagrama de classe de gerenciamento de menu..... | 45 |
| Figura 10 – Diagrama de classe de gerenciamento de conteúdo..... | 46 |
| Figura 11 – Diagrama de classe de gerenciamento de galeria de imagens..... | 47 |
| Figura 12 – Diagrama de classe de configuração do sistema | 48 |
| Figura 13 – Diagrama de seqüência da adição de menu..... | 49 |
| Figura 14 – Diagrama de seqüência da adição de conteúdo..... | 50 |
| Figura 15 – Diagrama de seqüência da adição de galeria de imagem..... | 51 |
| Figura 16 – Arquivo XML sendo montado utilizando PHP | 54 |
| Figura 17 – Arquivo de <i>template</i> sem a substituição das palavras reservadas | 55 |
| Figura 18 – Arquivo de <i>template</i> com a substituição das palavras reservadas | 56 |
| Figura 19 – Autenticação do usuário | 57 |
| Figura 20 – Tela principal do sistema | 58 |
| Figura 21 – Tela de configuração do sistema | 59 |
| Figura 22 – Sistema confirmando sucesso da operação | 60 |
| Figura 23 – Gerenciamento de menu..... | 61 |

| | |
|--|----|
| Figura 24 – Inserção de um novo item de menu..... | 62 |
| Figura 25 – Gerenciamento de conteúdo com um item de menu selecionado | 63 |
| Figura 26 – Inserção de um novo conteúdo..... | 64 |
| Figura 27 – Gerenciamento da Galeria de Imagens | 65 |
| Figura 28 – Inserção de uma imagem na galeria | 66 |
| Figura 29 – Escolha do arquivo a ser enviado..... | 67 |
| Figura 30 – Barra de progresso no envio do arquivo | 68 |
| Figura 31 – Visualização do <i>website</i> com os conteúdos cadastrados..... | 69 |
| Quadro 9 – Checklist ErgoList e seus respectivos itens..... | 73 |

LISTA DE SIGLAS

AS3 – *ActionScript 3*

ASP – *Active Server Pages*

AVM – *ActionScript Virtual Machine*

CMS – *Content Management System*

FTP – *File Transfer Protocol*

HTML – *Hypertext Markup Language*

HTTP – *Hypertext Transfer Protocol*

JIT – *Just in Time*

JSP – *JavaServer Pages*

PHP – *Hypertext Preprocessor*

RIA – *Rich Internet Application*

WYSIWYG – *What You See Is What You Get*

MXML – *Macromedia Flex Markup Language*

JIT – *Just in Time*

SGML – *Standard Generalized Markup Language*

XML – *Extensible Markup Language*

UML – *Unified Modeling Language*

UTF-8 – *8-bit UCS/Unicode Transformation Format*

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO..... | 12 |
| 1.1 OBJETIVOS DO TRABALHO | 13 |
| 1.2 ESTRUTURA DO TRABALHO | 13 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 15 |
| 2.1 SISTEMAS DE GERENCIAMENTO DE CONTEÚDO..... | 15 |
| 2.2 RIA | 17 |
| 2.3 FLEX | 19 |
| 2.3.1 Adobe Flash Player | 23 |
| 2.3.2 ActionScript 3 | 24 |
| 2.4 XML | 25 |
| 2.5 PHP..... | 25 |
| 2.6 ERGOLIST | 26 |
| 2.7 TRABALHOS CORRELATOS..... | 27 |
| 2.7.1 Yahoo Maps | 27 |
| 2.7.2 Google Earth | 28 |
| 2.7.3 JOOMLA | 28 |
| 2.7.4 Mambo Server..... | 29 |
| 3 DESENVOLVIMENTO DO TRABALHO..... | 30 |
| 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO..... | 30 |
| 3.2 ESPECIFICAÇÃO | 31 |
| 3.2.1 Diagramas de Caso de Uso | 31 |
| 3.2.2 Diagrama de Atividades..... | 41 |
| 3.2.3 Diagrama de Entidade e Relacionamento – DER | 42 |
| 3.2.4 Diagramas de Classes..... | 43 |
| 3.2.4.1 Diagrama de classe: Gerenciamento de Menu..... | 45 |
| 3.2.4.2 Diagrama de classe: Gerenciamento de Conteúdo | 46 |
| 3.2.4.3 Diagrama de classe: Gerenciamento de Galeria de Imagens..... | 47 |
| 3.2.4.4 Diagrama de classe: Configuração do Sistema..... | 48 |
| 3.2.5 Diagramas de Seqüência | 48 |
| 3.2.5.1 Diagrama de seqüência: Adição de Menu | 49 |
| 3.2.5.2 Diagrama de seqüência: Adição de Conteúdo | 50 |

| | |
|---|-----------|
| 3.2.5.3 Diagrama de seqüência: Adição de Galeria de Imagem..... | 51 |
| 3.3 IMPLEMENTAÇÃO | 52 |
| 3.3.1 Técnicas e ferramentas utilizadas..... | 52 |
| 3.3.1.1 phpMyAdmin..... | 52 |
| 3.3.1.2 FLEX | 53 |
| 3.3.1.3 Dreamweaver | 53 |
| 3.3.1.4 ErgoList | 54 |
| 3.3.1.5 Templates..... | 55 |
| 3.3.2 Operacionalidade da implementação | 56 |
| 3.4 RESULTADOS E DISCUSSÃO | 69 |
| 4 CONCLUSÕES..... | 74 |
| 4.1 EXTENSÕES | 75 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 76 |

1 INTRODUÇÃO

No terceiro trimestre de 2005, segundo o DOSSIÉ... (2005, p. 2), haviam 85,6 milhões de nomes de domínio em todo o mundo. Este número mostra um crescimento de 3% comparado com o segundo trimestre de 2005, e um crescimento de 29% no ano. O crescimento recorde na criação de domínios na internet mostra que o interesse das pessoas e empresas em participar, mostrar o seu produto ou trazer informações diversas é cada vez maior. Não é por acaso que a internet é considerada uma das maiores fontes de informação e de pesquisa existentes, bem como uma ótima vitrine e fonte de renda para pessoas físicas e jurídicas.

De acordo com Cardozo (2006, p. 78), “Hoje em dia, mesmo empresas pequenas têm a necessidade de manter um site dinâmico, a fim de guardar suas informações em um banco de dados. Isto para não falar das maiores que utilizam intranets [...]”. Esta necessidade de possuir *websites* dinâmicos, publicar artigos, fotos, vídeos e informações, requer, de maneira geral, que os usuários possuam certo conhecimento avançado em informática para gerenciar os conteúdos na internet.

Tentando solucionar este problema da constante necessidade de atualizações nos *websites*, os Sistemas de Gerenciamento de Conteúdo, ou *Content Management Systems* (CMS), vieram para facilitar e auxiliar na manutenção dos conteúdos na internet, reduzindo o tempo gasto nestas atualizações e possibilitando que os usuários não necessitem de conhecimentos avançados em informática. Os CMSs permitem a criação, edição ou exclusão de conteúdos de forma segura e consistente, apresentando todas as informações para o usuário de forma que o mesmo tenha total controle sob suas ações.

Neste trabalho é desenvolvido um aplicativo web para permitir o gerenciamento de conteúdos de páginas na internet. O ambiente de programação utilizado é o FLEX, para que permita o gerenciamento destes conteúdos em tempo real e também para possibilitar que a aplicação tenha as funcionalidades e velocidade de resposta típica de aplicativos *desktop*, evitando assim um dos inconvenientes de se desenvolver aplicações para web, que segundo McLellan (2005), qualquer interação numa aplicação web tradicional requer o completo envio da página para o servidor para que ele possa reconstruí-la e retornar ao cliente.

O sistema desenvolvido também faz uso da linguagem de programação *Hypertext Preprocessor* (PHP) para fazer a comunicação com o banco de dados MySQL e retornar o resultado dessa comunicação para o sistema utilizando a linguagem de marcação genérica,

eXtensible Markup Language (XML), além de apresentar algumas características de Rich Internet Application (RIA), como a capacidade de processar e executar código diretamente no cliente sem a necessidade de fazer comunicação com o servidor, possuir transparência na comunicação com servidores de banco de dados e a possibilidade de executar em múltiplas plataformas. Também é utilizado o ErgoList (CYBIS, 1998) para resolver as questões relacionadas a usabilidade e melhorias na interface do sistema para facilitar o uso de usuários sem maiores conhecimentos em informática.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema de gerenciamento de conteúdo para a web utilizando FLEX.

Os objetivos específicos do trabalho são:

- a) desenvolver uma aplicação web para gerenciamento de conteúdo de um *website* de uma empresa;
- b) permitir que os usuários possam inserir, editar e excluir conteúdos no *website* com formatação de texto utilizando recursos de como negrito, itálico e sublinhado, além do envio de imagens de forma segura e consistente sem a necessidade de conhecimento de programação;
- c) disponibilizar no *website* o conteúdo armazenado.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos.

No primeiro capítulo, é apresentada, uma introdução ao trabalho, seus objetivos e sua estrutura a fim de fornecer as informações necessárias para o entendimento do assunto e projeto a ser desenvolvido.

No segundo capítulo, é fornecida uma explanação sobre alguns conceitos de aplicações web, além de fornecer uma visão geral sobre sistemas de gerenciamento de conteúdo e RIA, além de alguns fundamentos que servem como base para o trabalho como o FLEX,

ActionScript 3, PHP e o ErgoList.

O terceiro capítulo descreve o desenvolvimento do trabalho, apresentando sua especificação com diagramas de caso de uso e diagramas de classe, bem como questões referentes a implementação.

O quarto capítulo apresenta as conclusões do trabalho, bem como suas atuais limitações e sugestões para continuação.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados, de forma resumida, conceitos técnicos sobre as principais áreas envolvidas no desenvolvimento deste trabalho como sistemas de gerenciamento de conteúdo, RIA, FLEX, ActionScript 3, PHP e o ErgoList. Também são apresentados alguns trabalhos correlatos.

2.1 SISTEMAS DE GERENCIAMENTO DE CONTEÚDO

O crescimento apontado pelo DOSSIÉ... (2005), indicando aumento de registros de domínios na internet mostra claramente objetivo das pessoas em mostrar o seu produto e alcançar cada vez mais um público maior. Por consequência, surge o desafio de fornecer serviços de qualidade para uma gama muito maior de pessoas, que a cada ano são mais exigentes, mais críticos e mais dependentes destes serviços na internet.

Sistemas de gerenciamento de conteúdo (CMS), segundo Synodinos (2005), são sistemas utilizados para facilitar a elaboração e criação de conteúdo digital, que com o apoio de grandes empresas, estes sistemas vão se tornar uma solução viável para empresas organizarem e darem a devida manutenção em seus respectivos *websites*. Já Browning e Lowndes (2001), acreditam que CMS está mais para um conceito que engloba uma série de processos específicos de cada empresa do que um produto em si.

No entanto, conforme Patrick (2006), sistemas de gerenciamento de conteúdo são fundamentalmente relacionados ao processo de organizar o *website* com maior eficiência e rapidez para colocar conteúdo de qualidade *online* no menor tempo possível. Segundo ele, estes sistemas ajudam pessoas sem conhecimento técnico em informática a gerenciar páginas complexas, dinâmicas, com muitas fotos, textos, artigos e documentos mais facilmente que o processo manual.

A importância estratégica para os negócios de manter um site atualizado, bem informado, organizado e de fácil utilização é de vital importância para quem pretende expandir os seus negócios para a internet. É comum encontrar empresas na internet que possuem uma pessoa no cargo de *webmaster* responsável por redigir textos, aplicar a formatação correta, inserir as imagens, construir os *links* para o novo documento e colocá-lo

para funcionar no *website* atual. Entretanto, esse processo pode ficar saturado e a pessoa pode ficar suscetível a cometer erros.

Patrick (2006) exemplifica o processo de atualização do *website* de uma empresa fictícia quando um diretor informa a uma funcionária que ele gostaria de atualizar o *website* em questão com um documento para a imprensa sem utilizar um gerenciador de conteúdo:

- a) Maria redige o documento em um editor de texto;
- b) ela anexa o documento em um e-mail e encaminha para o *designer*;
- c) João recebe o documento e salva o anexo;
- d) ele abre o seu editor de HTML preferido na sessão principal do *website*;
- e) ele cria um *link* para este novo documento que lhe foi encaminhado além de inserir uma chamada na página principal para este documento;
- f) o documento é programado em HTML manualmente e adicionado ao *website*;
- g) João envia uma nota a Maria informando um endereço temporário para revisão do documento;
- h) Maria revisa o documento e faz suas anotações copiando e colando trechos do documento original para enviar para João;
- i) João recebe as atualizações por e-mail e atualiza o *website*;
- j) João envia outra nota a Maria informando que as alterações foram concluídas e estão prontas para revisão novamente;
- k) Maria faz uma revisão final no documento e informa João que tudo está pronto para ir para o *website* por definitivo;
- l) João recebe o email de Maria, move os arquivos que foram alterados e criados do *website* temporário para o *website* definitivo.

Exemplificando o mesmo caso com o uso de um sistema de gerenciamento de conteúdo, Patrick (2006) define os passos desta forma:

- a) Maria digita seu *login* e senha para acessar o sistema;
- b) usando o *link* de “Novo Documento”, Maria redige o documento formatando o texto conforme o necessário;
- c) Maria revisa o documento e faz as alterações necessárias;
- d) Maria pressiona o botão para publicar o documento e o *website* é atualizado com o novo conteúdo redigido e revisado.

A diferença de um processo para o outro é muito clara. Sem o gerenciador de conteúdo, o processo poderia ter levado dias dependendo da quantidade de serviço das pessoas envolvidas. Já com o sistema, a adição do conteúdo pode levar menos de 15 minutos.

De acordo com Millarch (2005), “Se a sua empresa é realmente ativa, tem constantemente novos clientes, produtos [...], é natural que você queira publicá-los em seu site. Se não o faz, perde oportunidades de negócios e de fortalecer sua imagem institucional”. Essa necessidade de organizar e atualizar este conteúdo, não é tarefa fácil e de acordo com o autor é comum encontrar empresas que contratam uma pessoa específica para fazer a atualização deste conteúdo. Este modelo tradicional de gerenciamento de conteúdo em *websites* não é ideal e as desvantagens deste modelo são citadas por Millarch (2005) como:

- a) administração atrelada a um computador com o software necessário para fazer as alterações, inclusões e exclusões do conteúdo;
- b) modificações acidentais são comuns, resultando em *links* quebrados, problemas de formatação, alterações no *layout* e até mesmo indisponibilidade do *website*;
- c) dificuldade em implementar um mecanismo de pesquisa granular e organizar as informações em seções e categorias para serem encontradas com maior facilidade.

2.2 RIA

De acordo com Eichorn (2006, p. 8), a internet evoluiu muito desde a sua criação. Tudo começou com simples comunicação de texto e hoje já é possível criar apresentações multimídia e grandes aplicações, contanto essas aplicações possuem seus problemas. A maioria destas aplicações são incrivelmente complicadas e desajeitadas quando comparadas com as mesmas aplicações nativas executadas nos sistemas operacionais, além de possuírem problemas de usabilidade.

As aplicações na internet trazem muitos benefícios em comparação com os aplicativos normais. São de fácil acesso, não requerem instalação, podem ser atualizados a qualquer momento e oferecem acesso a grandes quantidades de dados sem a necessidade de redes complexas. Estas vantagens reduzem o tempo de desenvolvimento e também os custos de manutenção dos aplicativos. Apesar dos aplicativos na internet possuírem interfaces de qualidade inferior devido a sua simplicidade e demora na atualização do conteúdo, estes aplicativos estão substituindo aplicativos nativos por toda parte.

Crane e Pascarello (2005, p. 17) mencionam que nas aplicações clássicas na web, o navegador não passa de um terminal burro. As informações são salvas no servidor, e toda vez que um usuário interage com um *website*, outro documento é lido e enviado para o navegador

contendo as informações de interface e dados requisitados. O navegador “joga” o documento anterior fora e mostra o novo.

Segundo Eichorn (2006, p. 8), Rich Internet Applications são aplicações na internet que tentam reduzir a distância de usabilidade entre as aplicações nativas e as aplicações da internet normais de hoje. Elas contêm mais código no navegador, que possuem maior interatividade e experiência similar a dos aplicativos nativos de um sistema operacional. As tecnologias existentes hoje para desempenhar o desenvolvimento destes aplicativos são Flash, FLEX, Java, ActiveX e JavaScript.

Já a Adobe Systems Incorporated (2006a), data o conceito de RIA como sendo introduzido em março de 2001 pela Macromedia, recentemente adquirida pela Adobe, baseando-se em conceitos que estavam na web por muitos anos. A idéia é fazer com que os usuários possam interagir nos seus navegadores, quebrando a famosa arquitetura cliente-servidor da web, arquitetura essa que faz com que toda comunicação deva passar por um servidor, que por sua vez necessita que os dados sejam enviados para então processá-los e devolver outro documento por completo. Utilizando tecnologias que podem ser executadas do lado do cliente, RIAs podem evitar este processo lento e repetitivo entre cliente e servidor.

De acordo com Rich Internet Application (2006), o conceito de RIA oferece algumas vantagens e desvantagens. As vantagens são:

- a) melhores interfaces: a interação do usuário utilizando estas tecnologias pode permitir que cálculos sejam feitos na própria página sem a necessidade do envio dos dados para o servidor, ou criar efeitos de *drag and drop* que não são possíveis somente com o uso do *HyperText Markup Language* (HTML);
- b) comunicação assíncrona: os aplicativos podem interagir com o servidor de forma assíncrona, não havendo a necessidade de esperar por uma interação do usuário com algum *link*.
- c) desempenho da rede: como algumas partes da lógica de programação pode ser executada no cliente, a necessidade de se enviar e receber toda a página não existe. Geralmente a troca de informações se restringe aos dados a serem apresentados para o usuário.

As desvantagens são:

- a) acesso restrito a memória: RIAs possuem acesso restrito a memória e recursos do sistema por questões de segurança. Se este acesso for desabilitado, estes aplicativos podem deixar de funcionar corretamente;
- b) *script* desabilitado: é comum os aplicativos na internet usarem JavaScript como

linguagem padrão para executar suas funções. No entanto, se o usuário estiver com o navegador configurado para não executar *scripts*, o aplicativo pode deixar de executar corretamente, ou não funcionar de forma nenhuma;

- c) *download* do *script*: é necessário que seja transmitido pelo servidor todo *script* de execução do aplicativo, o que, dependendo do tamanho do *script*, pode causar lentidão para carregar a página.

Tapper et al. (2006, p. 9) separam as vantagens em 3 grupos:

- a) empresas de negócios: fazendo com que a interatividade do usuário seja mais fácil, o número de transações completas aumenta em portais de venda de produtos por exemplo;
- b) empresas de tecnologia de Informação (TI): saindo do modelo tradicional de apresentar uma página de cada vez, nota-se a redução do uso dos servidores que hospedam as páginas, dos servidores de banco de dados e do tráfego de dados;
- c) usuários finais ou consumidores: os usuários finais são os que mais ganham neste processo todo. Uma aplicação nestes conceitos, reduz o índice de frustração do usuário porque eles não necessitam navegar por várias páginas para encontrar o que precisam ou ter que aguardar que todo conteúdo da página seja carregado para depois ser apresentado.

2.3 FLEX

De acordo com Adobe Systems Incorporated (2006a), muitos aplicativos ou páginas na web deixam a desejar, especialmente quando comparados com a riqueza e usabilidade dos melhores programas *desktop*.

Para os consumidores, páginas de e-commerce apresentadas no modelo tradicional cliente-servidor e a falta de inteligência na execução de códigos no navegador do usuário, podem fazer simples transações confusas e passíveis de erros, fazendo com que empresas percam milhões de dólares em função de carrinhos de compras abandonados ou caras ligações para serviços de atendimento ao consumidor.

Para empresas, o modelo de distribuição de aplicativos na web é muito menos custoso e muito mais prático, visto que não há a necessidade de se instalar o produto na máquina do usuário, visto que o aplicativo fica hospedado em um servidor e o usuário apenas acessa como

uma página na internet. Mas ainda se espera um retorno na usabilidade e eficácia dos aplicativos web equivalente a dos aplicativos desktop. Como resultado, são gastos milhões de dólares na instalação de aplicativos e requisitos necessários para executá-los todos os anos.

Segundo Tapper et al. (2006, p. 10), o FLEX foi criado para desenvolver RIAs, compilando aplicativos para executarem no Adobe Flash Player, permitindo que os desenvolvedores possam criar interfaces similares a dos aplicativos normais executados diretamente no computador, além dos benefícios dos aplicativos tradicionais existentes hoje na internet. Os aplicativos desenvolvidos em FLEX são similares aos aplicativos desenvolvidos em JavaScript utilizando AJAX, isto é, ambos são capazes de atualizações dinâmicas na interface com o usuário, bem como a possibilidade de enviar e receber dados em segundo plano.

Os aplicativos desenvolvidos em FLEX utilizam a linguagem *Macromedia Flex Markup Language* (MXML) para especificação da interface do aplicativo. MXML funciona de forma muito similar a HTML dos navegadores padrões. Para a especificação da lógica da aplicação, o FLEX utiliza a linguagem ActionScript 3, desta forma as camadas de interface do usuário e da lógica de programação ficam separadas umas das outras. O Quadro 1 mostra um exemplo da sintaxe de um arquivo MXML para montar a interface de um aplicativo.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
3   <mx:TabNavigator x="21" y="23" width="200" height="200">
4     <mx:Canvas label="Aba 1" width="100%" height="100%">
5       </mx:Canvas>
6     <mx:Canvas label="Aba 2" width="100%" height="100%">
7       </mx:Canvas>
8   </mx:TabNavigator>
9 </mx:Application>
```

Quadro 1 – Exemplo da sintaxe de um arquivo MXML

Este exemplo de sintaxe apresentado no Quadro 1 gera uma interface com abas muito similar as mesmas utilizadas nos navegadores mais modernos, como pode ser observado na Figura 1.

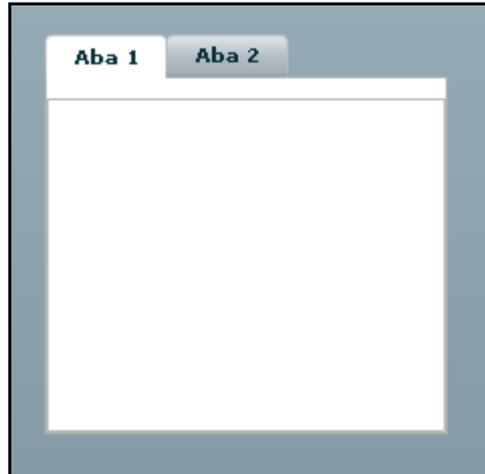


Figura 1 – Exemplo da tela gerada pelo exemplo de código fonte do Quadro 1.

Utilizando o FLEX para desenvolver RIAs, permite que os aplicativos desenvolvidos possam requisitar dados adicionais ou conteúdo através de requisições *HyperText Transfer Protocol* (HTTP) ou web services, funcionando em qualquer ambiente de servidor, podendo utilizar qualquer linguagem de programação voltada para a web como *JavaServer Pages* (JSP), *Active Server Pages* (ASP), PHP ou ColdFusion para fazer requisições ao banco de dado, ou utilizar o FLEX Data Services que permite fazer este tipo de requisição da mesma forma que é possível com estas linguagens. A única necessidade que se faz, é que o usuário possua o *plugin* Adobe Flash Player 9 instalado em seu navegador.

A Figura 2 mostra o fluxo de interação entre o navegador do usuário executando uma aplicação web compilada no FLEX e um servidor web com a indicação das transações feitas entre ambos.

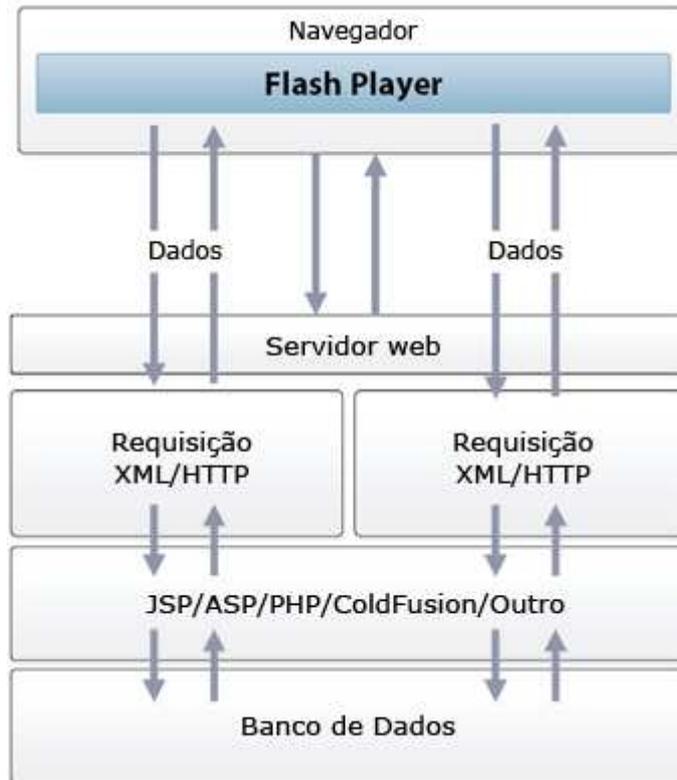


Figura 2 – Fluxo de interações entre aplicação *web* e servidor

Segundo Adobe Systems Incorporated (2006a), as vantagens oferecidas pelo FLEX em cima do desenvolvimento tradicional em HTML é o que faz com que ele se destaque no desenvolvimento de RIAs, algumas destas vantagens são:

- utilização de padrões: como o FLEX compila seus projetos em Flash, não existe mais o problema da falta de padronização que ocorre ao se desenvolver em linguagens nativas dos navegadores, como o JavaScript. O Flash permite executar seu código em múltiplas plataformas da mesma forma, seja Linux, Mac ou Windows;
- problemas de *layout*: utilizando o MXML para definir a interface, não ocorrem os problemas comuns de quem desenvolve em HTML como a falta de padronização na interpretação das *tags* por cada navegador;
- suporte a várias mídias: suporte a áudio, imagem e vídeo nativas, sem a necessidade de componentes adicionais;
- desempenho: os aplicativos são arquivos binários, assim como nos aplicativos *desktop*. Isto torna a execução do código muito mais rápida ao se comparar com as linguagens interpretadas em tempo real que são utilizadas hoje.

Outras vantagens também são mencionadas por Fain, Rasputnis e Tartakovsky (2006), estas vantagens são:

- a) o tamanho do *plugin* necessário é pequeno;
- b) as aplicações podem rodar fora do navegador quando feitas para tal função;
- c) oferece integração com todos os tipos de mídias de áudio e vídeo;
- d) oferece programação baseada em componentes, eliminando a programação de baixo nível.

Nas seções a seguir são apresentadas maiores informações sobre o Adobe Flash Player, *plugin* utilizado por aplicações desenvolvidas em FLEX e a linguagem de programação, ActionScript 3.

2.3.1 Adobe Flash Player

O Adobe Flash e o Adobe Flash Player referem-se a um programa usado para criar animações para a web, fugindo das tradicionais imagens estáticas. A idéia inicial é fazer um programa que possibilitasse ao usuário criar animações ou *websites* interativos utilizando vetores ou imagens através de uma ferramenta intuitiva e prática. De acordo com Adobe Systems Incorporated (2006b), o Flash Player é um dos *plugins* mais utilizados, sendo utilizado em mais de 97,3% dos computadores no mundo.

Desde as suas versões iniciais, o Flash vem recebendo grandes atualizações, visto que seu foco ampliou muito desde o conceito inicial. Nas primeiras versões, ele apenas fornecia animações simples de vetores baseados em uma linha de tempo com ações limitadas e simples, mas o produto foi atualizado ao longo dos anos e com isso aumentou sua capacidade e recursos.

Conforme Adobe Systems Incorporated (2006c), agora em sua versão mais recente, além de permitir a criação de *websites* animados e interativos, o Flash pode ser usado para o desenvolvimento de jogos para web, CDs de apresentação multimídia e desenvolvimento para dispositivos móveis como celulares, palmtops e smartphones com o Flash Lite. Em sua última versão, o Adobe Flash, e conseqüentemente o Adobe Flash Player, receberam uma grande reestruturação com a atualização da linguagem ActionScript para a versão 3, trazendo também uma nova máquina virtual e grandes melhorias de performance. Tapper et al. (2006, p. 10) ainda menciona a adição do *ActionScript Virtual Machine* (AVM) para melhorar o desempenho dos aplicativos, além de fornecer melhores ferramentas para depuração no desenvolvimento das aplicações.

2.3.2 ActionScript 3

ActionScript é a linguagem nativa utilizada para programação no FLEX e no Flash. O ActionScript 1.0 apareceu inicialmente no Flash 5.0, possuindo apenas algumas funções básicas e rotinas de controle da linha do tempo de animações criada no Flash, mas com as versões mais novas ela recebeu grandes adições, como suporte a orientação a objetos, novos tipos de dados e bibliotecas adicionais para cálculos e controle de animações.

De acordo com Adobe Systems Incorporated (2006a), o ActionScript 3 é a última atualização que a linguagem recebeu trazendo algumas novidades importantes. Uma delas, mencionada anteriormente, é a adição do AVM que permite execução mais rápida dos aplicativos. Outra adição feita nesta versão foi o compilador *Just in Time* (JIT), ou seja, o código compilado é convertido para código de máquina nativamente pelo Flash Player em tempo de execução, o que torna ele muito mais rápido na execução e mais eficiente no uso de memória que a linguagem JavaScript, por exemplo.

A linguagem segue agora os padrões ECMAScript, a mesma utilizada pela padronização da linguagem JavaScript. O Quadro 2 mostra um exemplo de classe criada na nova versão do ActionScript.

```
1      package com.michel.testes
2      {
3          import mx.containers.Accordion;
4          import mx.core.Container;
5          import flash.events.Event;
6
7          public class AccordionIcons extends Accordion
8          {
9              [Embed(source="upIcon.gif")]
10             [Bindable]
11             public var upIcon:Class;
12             public var selectedIcon:Class;
13             private var defaultWidth:int = 400;
14             private var defaultHeight:int = 300;
15
16             public function AccordionIcons(){
17                 super();
18                 // set defaultWidth and defaultHeight
19                 this.width = this.defaultWidth;
20                 this.height = this.defaultHeight;
21             }
22         }
```

Quadro 2 – Exemplo de um trecho de definição de uma classe em ActionScript 3

2.4 XML

XML é uma linguagem de marcação com sintaxe muito similar ao HTML, com o propósito de facilitar o compartilhamento, ou troca, de dados entre aplicações diferentes. De acordo com Walsh (1998), XML não especifica semântica ou um conjunto de *tags*. Ele fornece facilidade em definir relações estruturais entre informações e dados.

Os documentos em XML podem ser categorizados entre bem formados e válidos. Os documentos bem formados são documentos que seguem os padrões do XML, com estrutura, elementos e atributos. Os que não seguem estes padrões, não são considerados XML. Documentos válidos são os que seguem um conjunto mais complexo e restrito de formatação do arquivo.

Originalmente formado de uma versão mais simplificada do *Standard Generalized Markup Language* (SGML), o XML logo tornou-se um padrão para todo tipo de troca de informações, armazenamento de informações em alguns casos e linguagens de *scripts* baseadas nele.

O XML é utilizado no desenvolvimento da aplicação como uma linguagem intermediária para troca das informações entre o PHP e o FLEX. As consultas requisitadas pelo FLEX para o PHP, são repassadas para o banco de dados MySQL e são retornadas no formato XML para a aplicação, possibilitando maior facilidade na apresentação e manipulação dos dados consultados dentro do FLEX.

2.5 PHP

De acordo com Converse e Park (2001), o PHP é uma linguagem de código aberto que executa no servidor, conhecido também como *server-side scripting*, possibilitando a criação de websites dinâmicos. Criado em 1994 por Rasmus Lerdorf, as primeiras versões do PHP foram utilizadas apenas em sua página pessoal, apenas em 1995 a primeira versão foi liberada para o uso de outras pessoas, e na época era chamado de Personal Home Page Tools.

Estima-se que em 1996 o PHP era utilizado por 15.000 sites e que em 1997 este número pulou para 50.000. Foi nesta época que o PHP deixou de ser um projeto de Rasmus e recebeu uma equipe mais organizada de desenvolvimento e teve seu código reescrito, dando

base para a versão 3 na época.

Em 1998 foi lançado a versão 4 com notáveis ganhos de performance onde utilizavam uma *engine* conhecida por Zend. Nesta data o PHP se tornou amplamente utilizado por vários desenvolvedores de *websites*, tendo constantes atualizações e adições em seu código. A mais recente versão, PHP 5, possui grandes melhorias, com destaques para a orientação a objetos, conexão com banco de dados e XML.

Os autores Converse e Park (2001), ainda mencionam algumas vantagens de se utilizar o PHP como:

- a) possuir licença gratuita;
- b) bom desempenho;
- c) boa segurança;
- d) código fonte aberto;
- e) fácil de usar;
- f) boa documentação.

2.6 ERGOLIST

De acordo com Nielsen (2003), usabilidade é um atributo de qualidade que informa o quão fácil uma interface é de se utilizar, bem como a palavra usabilidade pode se referenciar também aos métodos para melhorar a facilidade de uso de um produto ou software durante o processo de planejamento.

A usabilidade é importante para responder algumas perguntas relacionadas a um determinado produto, como por exemplo, informar a facilidade com que os usuários cumprem determinadas tarefas, qual o tempo estimado para se executar determinadas tarefas depois do usuário ter aprendido como manusear o produto e até mesmo saber o grau de satisfação relacionado ao uso da aplicação ou produto.

O ErgoList (CYBIS, 1998) é um checklist para avaliação de interfaces homem-computador disponibilizado via web e tem o objetivo de disponibilizar uma ferramenta para avaliação da facilidade de uso de softwares. Segundo Nielsen (2003), usabilidade é uma das condições necessárias para sobrevivência de um *website* ou sistema. Se algo for complicado de se utilizar, ler ou obter informações, as pessoas simplesmente deixam de usá-lo, já no caso de intranets isso se torna uma questão de produtividade. Segundo Cybis (1998), “Um

checklist para a avaliação ergonômica representa uma ferramenta especialmente adequada para que projetistas [...] possam conceber dispositivos de software corretos sob o ponto de vista da interação com o usuário”.

2.7 TRABALHOS CORRELATOS

Esta seção tem como objetivo apresentar alguns trabalhos correlatos. Alguns destes trabalhos utilizam tecnologias similares as tecnologias utilizada no desenvolvimento deste trabalho. Outros são sistemas de gerenciamento de conteúdo com algum tempo de desenvolvimento e presença no mercado.

2.7.1 Yahoo Maps

Um dos trabalhos que mais se destacam na utilização do FLEX e da aplicação de alguns conceitos de RIA hoje em dia é o *site* Yahoo Maps (YAHOO, 2006). O Yahoo Maps é uma aplicação desenvolvida para a web em que o usuário possui a visualização aérea de um mapa dos Estados Unidos e do resto do mundo. Nesta visualização se tem a possibilidade de procurar uma localidade qualquer, seja esta localidade o nome de uma rua, cidade ou estado. Após a procura, a visualização do mapa é focada no local do resultado da procura, permitindo a utilização de alguns recursos extras como *zoom* para aproximar o mapa ao ponto de possibilitar a visualização do nome e o traçado das ruas de quase todas as cidades, além da possibilidade de arrastar o mapa com o *mouse* para observar outras áreas do mapa, bem como a visualização de imagens tiradas por satélites.

Outro recurso que o *site* possui é a procura de dois ou mais lugares para traçar uma rota de viagem, por exemplo, ao digitar o ponto de origem e destino, o site traça a rota e mostra no mapa o caminho a ser percorrido informando o nome das ruas e qual direção deve ser tomada em cada uma delas. Sua única limitação atualmente é que detalhamento de ruas e rotas se restringe apenas ao país Estados Unidos.

2.7.2 Google Earth

Outro trabalho que se destaca na área de RIA é o Google Maps (GOOGLE, 2006). O Google Maps é muito similar ao Yahoo Maps (YAHOO, 2006), ambos fornecem praticamente as mesmas funcionalidades e apresentam características muito semelhantes. Ele apresenta uma visão aérea de um mapa, onde é possível pesquisar por uma localidade, mover o mapa, dar *zoom*, pegar direções de uma localidade para outra e ver o mapa com fotos de satélite. A diferença entre o Google Maps e o Yahoo Maps, é que o Google Maps foi desenvolvido em HTML e JavaScript, já o Yahoo Maps foi feito utilizando FLEX. Referente a funcionalidade, a principal diferença entre eles fica por conta da interface mais ágil e rápida do Yahoo Maps e do desempenho. Ao mandar o Google Maps traçar o caminho entre dois pontos e fazer a mesma procura no Yahoo Maps, observa-se que o Yahoo Maps vai retornar com o resultado muito rápido. Isto ocorre pelo fato dele ter sido feito em FLEX e utilizar um *plugin* com linguagem compilada e não linguagem de *script*.

2.7.3 JOOMLA

O Joomla (2006) é uma nova ferramenta de gerenciamento de conteúdo para a web, mas que não utiliza nenhum recurso do conceito de RIA. A ferramenta é programado em PHP, HTML e JavaScript, mas nem por isto deixa de ser inferior. De fato, de acordo com a avaliação feita por Cardozo (2006), ele foi o que melhor se destacou dentre os outros gerenciadores de conteúdos avaliados. Os outros gerenciadores avaliados por Cardozo (2006) foram o Postnuke (PNSF, 2001), Siteframe (CAMPBELL, 2005) e WCMS (GEORGI, 2002). Dentre estes, o Joomla se destacou por ser o mais robusto em termos de recursos, por possuir a melhor e mais amigável interface gráfica e por possuir uma comunidade de desenvolvedores numerosa e ativa. Outro destaque do sistema foi a possibilidade de instalação de temas, idiomas e módulos para extensão da funcionalidade do mesmo, somando um total de 400 extensões no momento.

2.7.4 Mambo Server

Já um trabalho com um pouco mais de tempo de desenvolvimento na área de gerenciamento de conteúdo é o Mambo Server (MAMBO, 2000). O Mambo Server é um sistema bem conhecido e difundido entre os gerenciadores de conteúdo. Além de ser um dos CMSs mais antigos existentes, ele se destaca por ser uma solução completa para a criação de um portal na internet permitindo todo o gerenciamento de conteúdo, usuários e restrição de acessos. Apesar de ser gratuito, ele fornece um suporte comercial que permite atualizações mais frequentes, corrigindo erros e trazendo novidades para o sistema. O Mambo Server também possui suporte a *templates de layout* para o portal de internet que possui e uma gama de recursos prontos e funcionais como sistema de busca, sistema de anúncios e *banners*, controle de estatísticas e fóruns.

3 DESENVOLVIMENTO DO TRABALHO

O software desenvolvido por este trabalho é um sistema de gerenciamento de conteúdo para *websites* que tem como finalidade facilitar a interação do usuário na manutenção destes conteúdos.

Este capítulo tem como objetivo detalhar as etapas de desenvolvimento do sistema de gerenciamento de conteúdo, que possui como primeiro tópico a descrição dos requisitos do problema a ser solucionado. Logo após, a segunda seção mostra a especificação do sistema através da apresentação de diagramas de casos de uso, atividades, classe e o diagrama de entidade e relacionamento (DER) para a estrutura das informações no banco de dados. A terceira seção descreve a implementação do sistema, técnicas, ferramentas utilizadas e a operacionalidade do mesmo, sendo a quarta e última seção destinada aos resultados obtidos e discussão relacionada aos trabalhos correlatos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos deste sistema de gerenciamento de conteúdo são detalhados abaixo e categorizados como Requisitos Não-Funcionais (RNF) e Requisitos Funcionais (RF):

- a) o sistema deve ser implementado utilizando o ambiente de programação FLEX (RNF);
- b) o sistema deve ser implementado utilizando a linguagem de programação Action Script 3 (RNF);
- c) o sistema será desenvolvido para a web (RNF);
- d) o sistema deverá utilizar um navegador de internet com suporte ao plugin Adobe Flash Player (RNF);
- e) o sistema não deve atualizar toda a interface a cada operação realizada (RNF);
- f) o sistema deverá seguir as recomendações de usabilidade do projeto ErgoList (CYBIS, 1998) (RNF);
- g) o sistema deverá ser executado em qualquer ambiente de servidor de hospedagem com suporte ao banco de dados MySQL e PHP (RNF);
- h) o sistema deverá possuir autenticação de usuários (RF);

- i) o sistema deverá permitir a inclusão, edição e exclusão de conteúdos (RF);
- j) o sistema deverá permitir o envio de imagens (RF);
- k) o sistema deverá permitir a formatação dos textos com opções de negrito, itálico, sublinhado e cor (RF).

3.2 ESPECIFICAÇÃO

Para representação da especificação deste trabalho foi escolhido o padrão *Unified Modeling Language* (UML), utilizando os diagramas de casos de uso, atividade e de classes. A ferramenta Enterprise Architect (SPARX SYSTEMS, 2007) foi escolhida para o desenvolvimento dos diagramas e foi utilizada em função de sua facilidade de uso e por dar suporte a todos os diagramas utilizados neste trabalho.

3.2.1 Diagramas de Caso de Uso

Para desenvolver o sistema foram identificados os principais casos de uso. Abaixo estão as figuras dos diagramas mostrando o cenário, seguido dos quadros explanando cada um deles.

A Figura 3 mostra o cenário de controle de acesso de um usuário ao sistema.

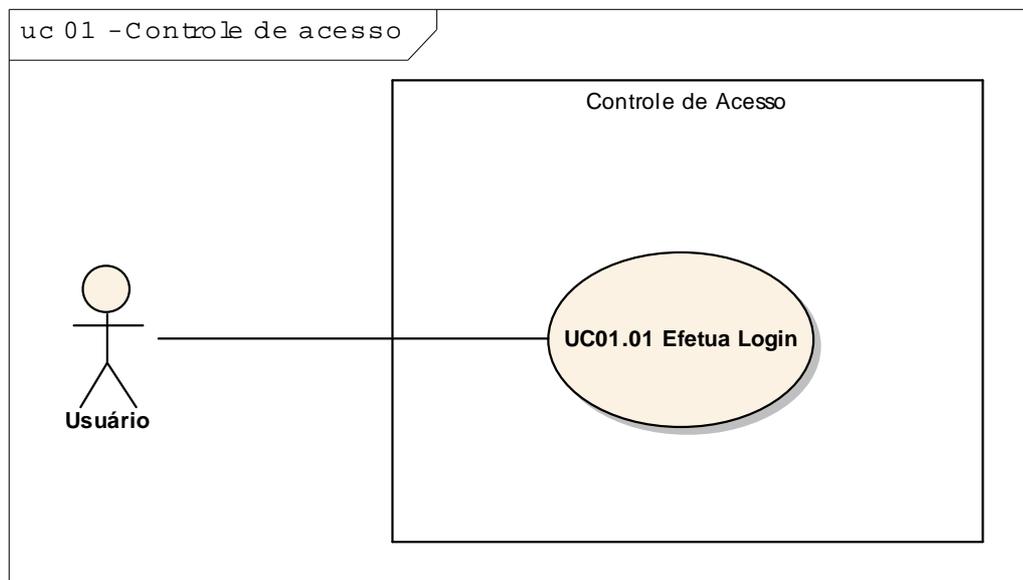


Figura 3 – Diagrama de caso de uso de controle de acesso

O Quadro 3 descreve o caso de uso efetua login, executado logo ao carregar o sistema.

UC01.01 Efetua Login

Descrição

Permite que o usuário possa ter acesso ao sistema.

Pré-condições

Não há.

Fluxo principal

1. O usuário informa o login e a senha.
2. O usuário confirma sua requisição clicando no botão de envio das alterações.
3. O aplicativo abre uma conexão com o banco de dados, valida a informação.
4. O aplicativo apresenta a tela principal do sistema.

Fluxo alternativo

Não há.

Fluxo de exceção

Erro na transmissão:

No passo 3, caso não seja possível transmitir os dados, é emitida a mensagem “Servidor não disponível ou falha na transmissão”.

Campos não preenchidos:

No passo 3, caso os campos obrigatórios não tenham sido preenchidos, é emitida uma mensagem requerendo o preenchimento dos mesmos.

Login ou senha inválidos:

No passo 3, caso as informações digitadas não forem válidas, é emitida a mensagem “Login/Senha inválidos. Por favor, digite corretamente as informações”.

Pós-condições

É apresentada a tela principal do sistema.

Requisitos atendidos

1. O sistema deverá possuir autenticação de usuários.

Quadro 3 – Caso de uso UC01.01 Efetua Login

A Figura 4 mostra o cenário geral do sistema apresentando as principais funcionalidades que o mesmo permite.

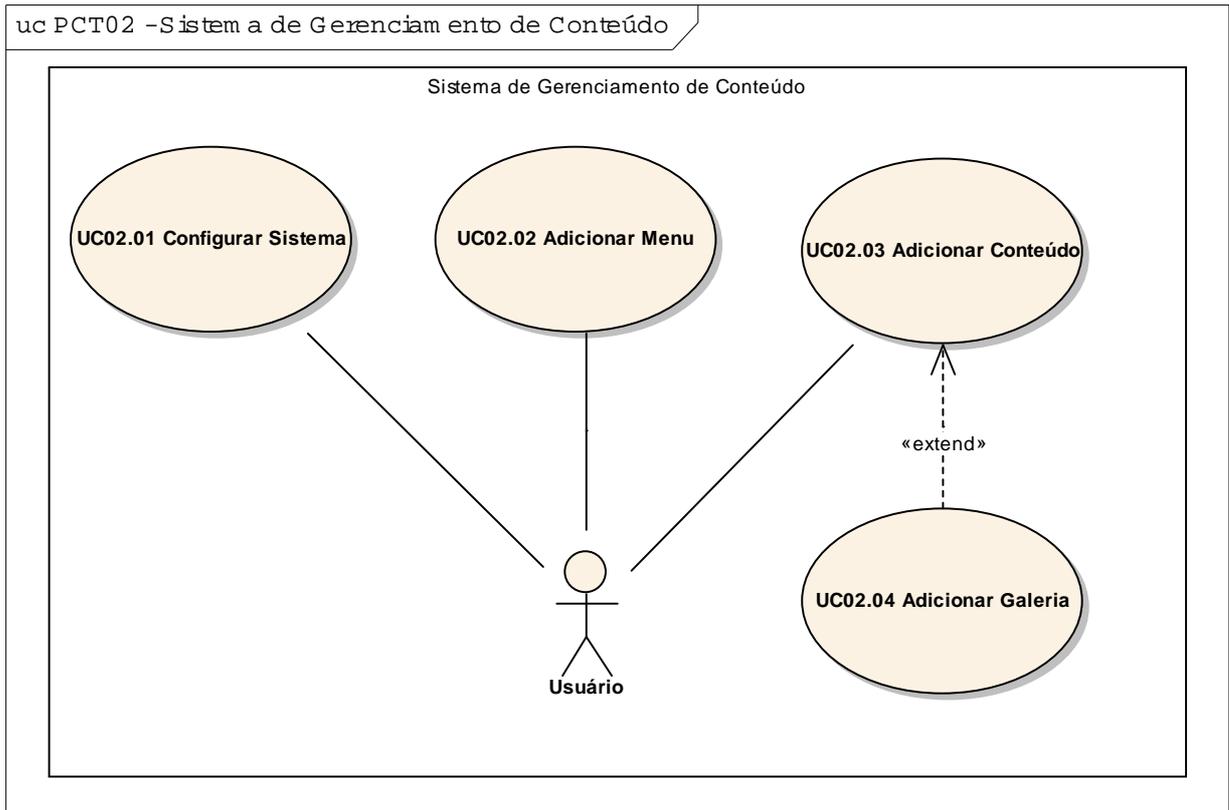


Figura 4 – Diagrama de casos de usos do sistema

O Quadro 4 descreve o caso de uso configurar sistema, executado quando o usuário deseja alterar alguma informação da *website*, como título, texto de direitos autorais ou até mesmo colocar o *website* em estado de manutenção, não permitindo acesso a página.

UC02.01 Configurar Sistema

Descrição

Permite acesso as funções de configuração do sistema de gerenciamento de conteúdo e do *website*.

Pré-condições

1. Usuário deve estar autenticado no sistema.

Fluxo principal

1. O aplicativo apresenta os campos de configuração do aplicativo e do *website*.
2. O usuário altera o login e senha de acesso, título, tema da aparência do site, *meta tags* de descrição e palavras chave para indexação dos sistemas de busca, texto dos direitos autorais e o estado do site caso esteja em manutenção ou não.
2. O aplicativo valida estas informações alteradas e abre uma conexão com o banco de dados para salvar estas alterações.
3. O aplicativo volta ao passo 1.

Fluxo alternativo

Não há.

Fluxo de exceção

Erro na transmissão:

No passo 2, caso não seja possível transmitir os dados, é emitida a mensagem “Servidor não disponível ou falha na transmissão”.

Campos não preenchidos:

No passo 2, caso os campos obrigatórios não tenham sido preenchidos, é emitida uma mensagem requerendo o preenchimento dos mesmos.

Pós-condições

As configurações do sistema são salvas no banco de dados.

Requisitos atendidos

Este caso de uso complementa o requisito abaixo, permitindo a alteração do *login* e senha do usuário.

1. Usuário deve estar autenticado no sistema.

Quadro 4 – Caso de uso UC02.01 Configurar Sistema

O Quadro 5 descreve o caso de uso adicionar menu, executado quando o usuário deseja adicionar, alterar ou excluir algum item de menu do *website*.

UC02.02 Adicionar Menu

Descrição

Permite a criação, alteração e exclusão de um item no menu.

Pré-condições

1. O usuário deve estar autenticado no sistema.

Fluxo principal

1. O aplicativo apresenta uma lista com os menus existentes.
2. O usuário opta por criar um novo menu.
3. O aplicativo apresenta os campos necessários para a criação de um novo item de menu.
4. O usuário informa nome e confirma sua requisição clicando no botão de envio das informações.
5. O aplicativo valida estas informações preenchidas e abre uma conexão com o servidor de banco de dados para salva-las.
6. O aplicativo volta ao passo 3.

Fluxo alternativo

Editar menu:

No passo 2, o usuário pode optar por editar um menu já existente.

- 2.1. O aplicativo apresenta os campos de um menu, já preenchidos com as informações do menu selecionado.
- 2.2. O usuário altera o nome, tipo de conteúdo e o estado deste menu e confirma sua requisição clicando no botão de envio das informações.
- 2.3. O aplicativo volta ao passo 5.

Excluir conteúdo:

No passo 2, o usuário pode optar por excluir um conteúdo já existente.

- 2.1. O aplicativo emite a mensagem “Deseja realmente excluir este item?” e aguarda confirmação do usuário.
- 2.2. O usuário confirma a exclusão.
- 2.3. O aplicativo abre uma conexão com o servidor de banco de dados e exclui o menu

desejado.

2.4. O aplicativo retorna ao passo 1.

Fluxo de exceção

Erro na transmissão:

No passo 5 e 2.3, caso não seja possível transmitir os dados, é emitida a mensagem “Servidor não disponível ou falha na transmissão”.

Campos não preenchidos:

No passo 5, caso os campos obrigatórios não tenham sido preenchidos, é emitida uma mensagem requerendo o preenchimento dos mesmos.

Pós-condições

Um novo item de menu é cadastrado, alterado ou excluído do aplicativo.

Requisitos atendidos

Este caso de uso complementa o requisito abaixo, permitindo a criação, alteração ou exclusão de um item de menu.

1. O sistema deverá permitir a inclusão, edição e exclusão de conteúdos.

Quadro 5 – Caso de uso UC02.02 Adicionar Menu

O Quadro 6 descreve o caso de uso adicionar conteúdo, responsável pela adição, exclusão ou alteração de conteúdos dos menus já cadastrados no *website*.

UC02.03 Adicionar Conteúdo

Descrição

Permite a criação, alteração e exclusão de conteúdo para um determinado menu.

Pré-condições

1. O usuário deve estar autenticado no sistema.
2. Possuir pelo menos um item de menu cadastrado no sistema.

Fluxo principal

1. O aplicativo apresenta uma lista com os menus existentes.

2. O usuário seleciona o menu que deseja alterar o conteúdo.
3. O aplicativo apresenta uma lista dos conteúdos pertencentes ao item de menu selecionado.
4. O usuário opta por criar um novo conteúdo.
5. O aplicativo apresenta os campos necessários para a criação de um novo conteúdo.
6. O usuário informa título, subtítulo, chamada, texto e seleciona a opção galeria de imagens caso queira ou não e confirma sua requisição clicando no botão de envio das informações.
7. O aplicativo valida estas informações preenchidas e abre uma conexão com o servidor de banco de dados para salva-las.
8. O aplicativo volta ao passo 3.

Fluxo alternativo

Editar conteúdo:

No passo 4, o usuário pode optar por editar um conteúdo já existente.

- 4.1. O aplicativo apresenta os campos da edição de um conteúdo já preenchidos com as informações do conteúdo selecionado.
- 4.2. O usuário altera o título, subtítulo, chamada, texto, seleciona a opção galeria de imagens para adicionar ou remover uma galeria de imagens e confirma sua requisição clicando no botão de envio das informações.
- 4.3. O aplicativo volta ao passo 7.

Excluir conteúdo:

No passo 4, o usuário pode optar por excluir um conteúdo já existente.

- 4.1. O aplicativo emite a mensagem “Deseja realmente excluir este item?” e aguarda confirmação do usuário.
- 4.2. O usuário confirma a exclusão.
- 4.3. O aplicativo abre uma conexão com o servidor de banco de dados e exclui o conteúdo desejado.
- 4.4. O aplicativo retorna ao passo 3.

Adição de galeria de imagens:

No passo 6, caso o usuário tenha optado por adicionar uma galeria de imagens, executa UC02.04.

Fluxo de exceção

Erro na transmissão:

No passo 7 e 4.3, caso não seja possível transmitir os dados, é emitida a mensagem “Servidor não disponível ou falha na transmissão”.

Campos não preenchidos:

No passo 7, caso os campos obrigatórios não tenham sido preenchidos, é emitida uma mensagem requerendo o preenchimento dos mesmos.

Pós-condições

Um conteúdo é cadastrado, alterado ou excluído do aplicativo.

Requisitos atendidos

1. O sistema deverá permitir a inclusão, edição e exclusão de conteúdos.
2. O sistema deverá permitir a formatação dos textos com opções de negrito, itálico, sublinhado e cor.

Quadro 6 – Caso de uso UC02.03 Adicionar Conteúdo

O Quadro 7 descreve o caso de uso adicionar galeria, executado quando o usuário deseja adicionar, alterar ou excluir uma galeria de imagens para algum determinado conteúdo.

UC02.04 Adicionar Galeria

Descrição

Permite a adição de fotos para uma galeria de imagens de um determinado conteúdo.

Pré-condições

1. O usuário deve estar autenticado no sistema.
2. Possuir pelo menos um conteúdo cadastrado no sistema.

Fluxo principal

1. O aplicativo apresenta uma lista com as legendas das fotos existentes.
2. O usuário opta por adicionar uma nova foto.
3. O aplicativo apresenta os campos necessários para a adição de uma nova foto.
4. O usuário seleciona a imagem que deseja enviar para o sistema, preenche a legenda da

imagem e confirma o envio clicando no botão de envio das informações.

5. O sistema recebe a imagem, abre uma conexão com banco de dados e salva as informações da foto.

6. Retorna ao passo 1.

Fluxo alternativo

Editar foto:

No passo 2, o usuário pode optar por editar uma foto já existente.

2.1. O aplicativo apresenta os campos necessários para a edição da foto, já preenchidos com as informações da imagem.

2.2. O usuário altera a legenda, seleciona uma nova imagem caso deseje substituí-la e confirma sua requisição clicando no botão de envio das informações.

2.3. O aplicativo volta ao passo 5.

Excluir foto:

No passo 2, o usuário pode optar por excluir uma foto já existente.

2.1. O aplicativo emite a mensagem “Deseja realmente excluir este item?” e aguarda confirmação do usuário.

2.2. O usuário confirma a exclusão.

2.3. O aplicativo abre uma conexão com o servidor de banco de dados e exclui o conteúdo desejado.

2.4. O aplicativo retorna ao passo 1.

Fluxo de exceção

Erro na transmissão:

No passo 3 e 2.3, caso não seja possível transmitir os dados, é emitida a mensagem “Servidor não disponível ou falha na transmissão”.

Campos não preenchidos:

No passo 3 e 2.3, caso os campos obrigatórios não tenham sido preenchidos, é emitida uma mensagem requerendo o preenchimento dos mesmos.

Pós-condições

Uma galeria de imagens é cadastrada, alterada ou excluída do aplicativo.

Requisitos atendidos

1. O sistema deverá permitir o envio de imagens.

Quadro 7 – Caso de uso UC02.04 Adicionar Galeria

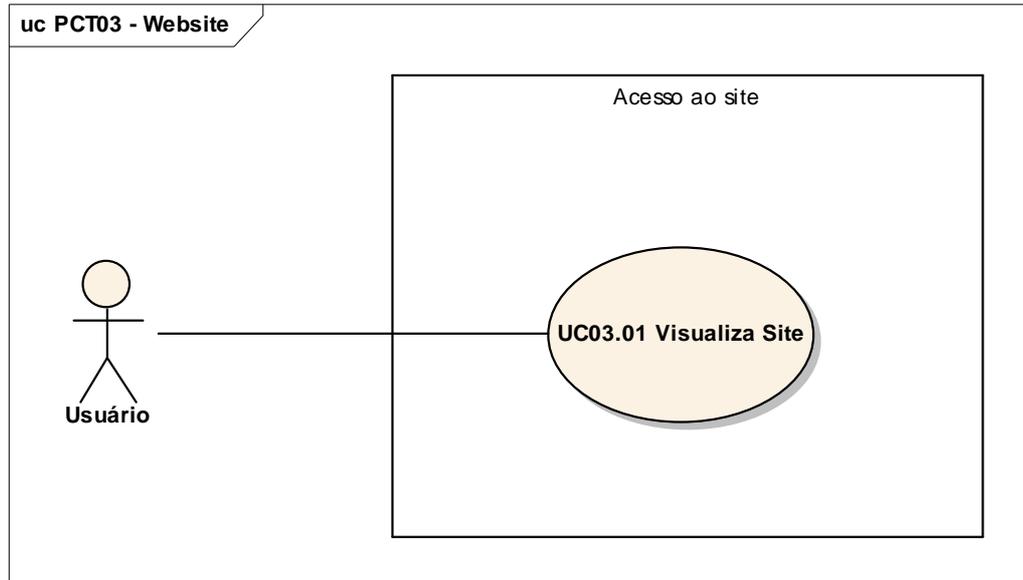


Figura 5 – Diagrama de casos de uso do *website*

O Quadro 8 apresenta o caso de uso visualiza site, executado quando o usuário acessa o *website* pelo qual o sistema de gerenciamento de conteúdo é responsável por manter.

UC03.01 Visualiza Site

Descrição

Permite a visualização do *website* gerenciado pelo sistema.

Pré-condições

1. Possuir pelo menos um menu com conteúdo cadastrado no sistema.

Fluxo principal

1. O aplicativo consulta no banco de dados as configurações, itens de menu e conteúdo do *website*.
2. O aplicativo carrega o *template* responsável pela aparência do *website*.
3. O aplicativo apresenta o *website* conforme todo conteúdo cadastrado.

Fluxo alternativo

Manutenção:

No passo 1, o caso o *website* esteja em manutenção.

2.1. O aplicativo apresenta uma mensagem informando que o *website* está em manutenção.

Fluxo de exceção

Não há.

Pós-condições

O *website* é apresentado para o usuário.

Requisitos atendidos

Este caso de uso complementa os requisitos abaixo, permitindo a visualização das imagens inseridas e dos conteúdos cadastrados com a formatação desejada.

1. O sistema deverá permitir a inclusão, edição e exclusão de conteúdos.
2. O sistema deverá permitir a formatação dos textos com opções de negrito, itálico, sublinhado e cor.
3. O sistema deverá permitir o envio de imagens.

Quadro 8 – Caso de uso UC03.01 Visualiza Site

3.2.2 Diagrama de Atividades

A Figura 6 apresenta o diagrama de atividades do sistema mostrando os passos seguidos para adicionar conteúdo no sistema até o acesso ao *website*.

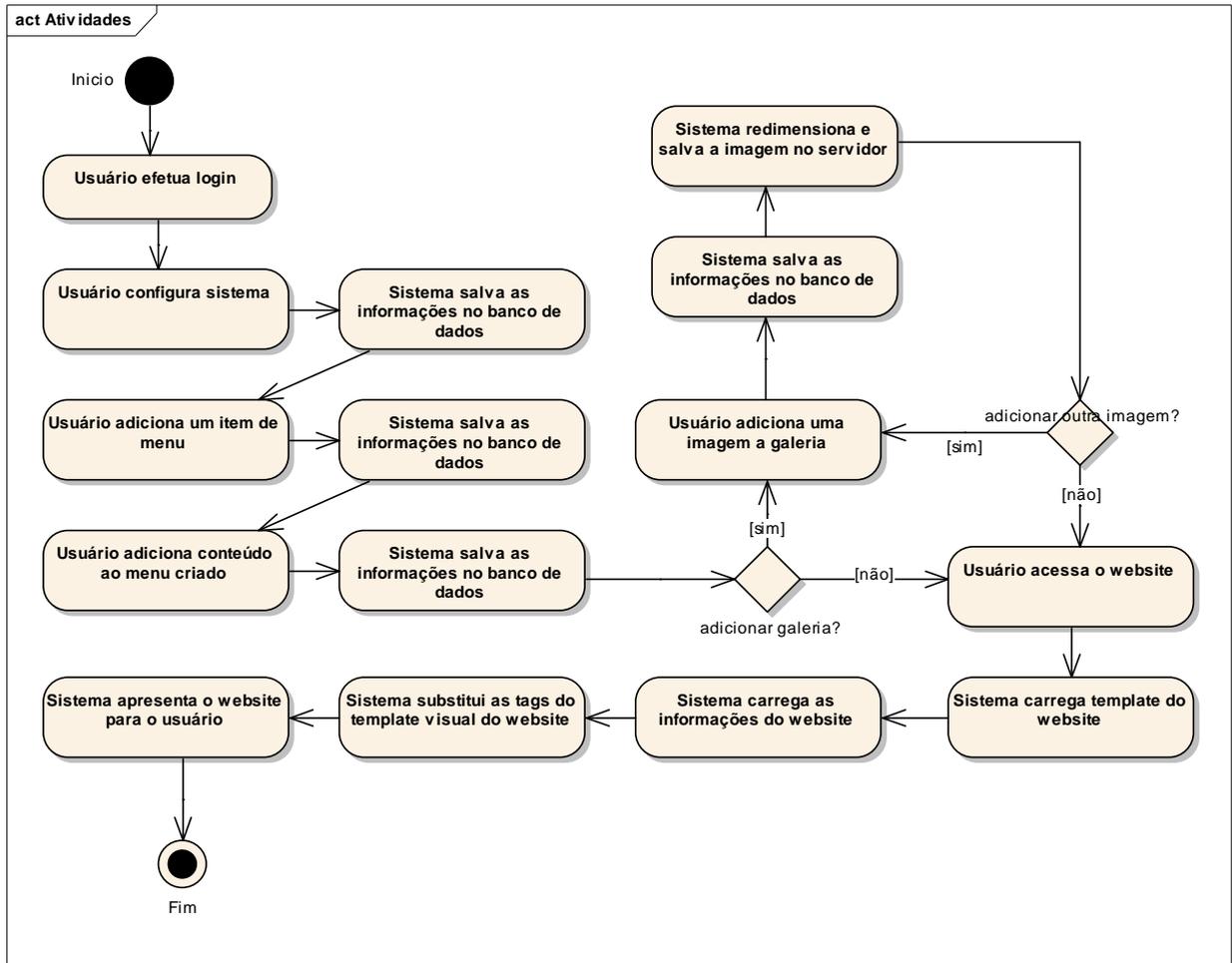


Figura 6 – Diagrama de atividades do sistema

3.2.3 Diagrama de Entidade e Relacionamento – DER

Os dados e informações utilizados pelo sistema são armazenados em um banco de dados MySQL. A Figura 7 destaca o DER desta base de dados apresentando as ligações entre as tabelas, além de mostrar como essa informação é armazenada indicando as variáveis envolvidas.

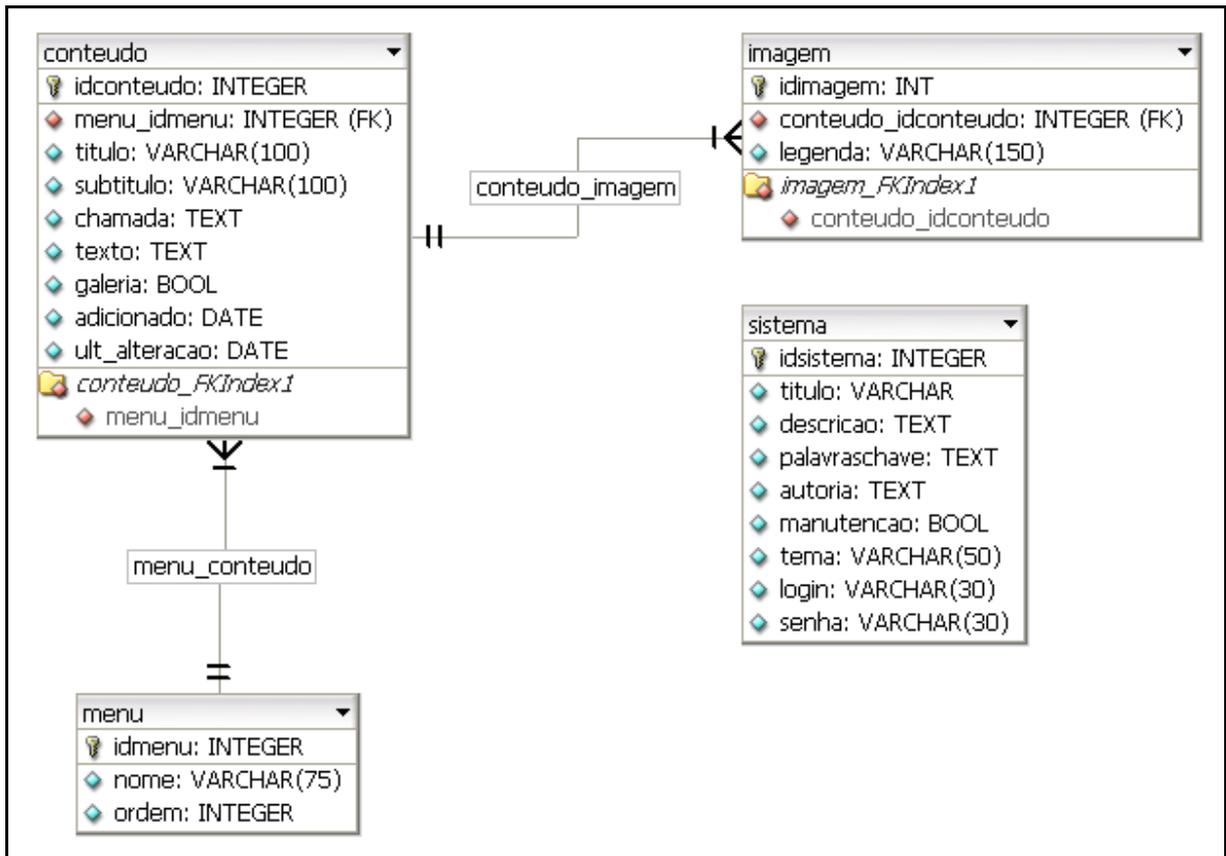


Figura 7 – Diagrama de entidade e relacionamento

3.2.4 Diagramas de Classes

O diagrama de classes é uma representação das classes utilizadas em sistema programado utilizando a orientação a objetos. Nestes diagramas são apresentadas as classes representando o conteúdo a ser manipulado sistema e as classes responsáveis pela interface do sistema seguindo os padrões da UML.

O FLEX utiliza os arquivos MXML para definir a formatação e apresentação da interface para o usuário, no entanto, estes arquivos também são utilizados como classes. Desta forma, a representação das classes que servem como interface para o usuário são apresentadas com um ícone no canto superior direito.

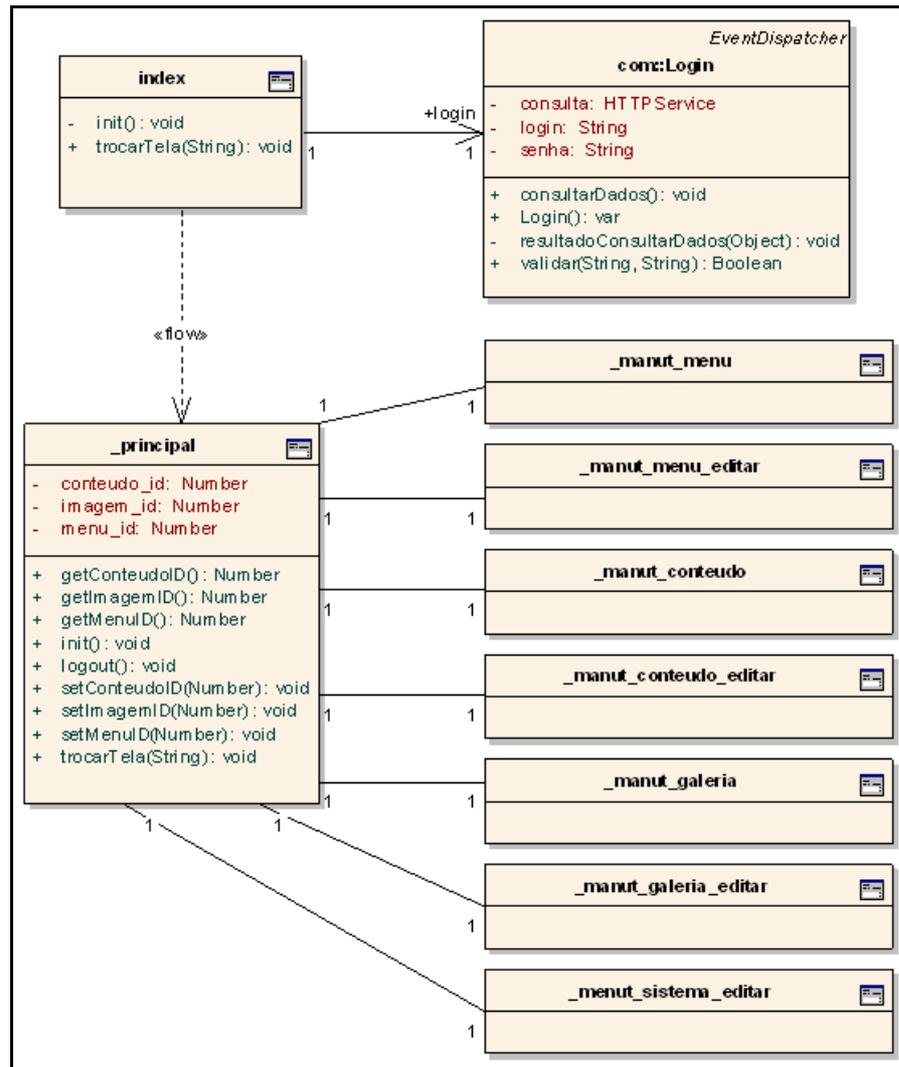


Figura 8 – Diagrama de classe da interface inicial

No caso da Figura 8, ela apresenta o diagrama de classes da interface inicial do sistema. Algumas das classes não são apresentadas com seus respectivos atributos e métodos, pois são detalhadas nas seções a seguir.

A classe construtora do sistema é a classe `index` que é responsável por apresentar a tela de autenticação ao sistema para usuário. A classe `login` é responsável por fazer a consulta ao banco de dados através um arquivo PHP que retorna o `login` e a senha para acesso ao sistema. Após receber estes dados ela dispara um evento, cujo qual é capturado pela classe `index`, que por sua vez libera os campos na interface principal para que eles sejam preenchidos.

Após a verificação de `login` e senha, é apresentada a interface principal do sistema com acesso aos recursos de gerenciamento de menu, conteúdo e configurações do sistema. A classe responsável por este papel é a classe `_principal`, que controla todas as telas apresentadas para o usuário. Qualquer chamada de outra classe solicitando a alteração da tela

atual do sistema para outra, é feita por esta classe.

Todas as consultas feitas pelo sistema ao banco de dados, seja para inserção, edição ou exclusão, passam por um arquivo PHP que consulta no banco de dados MySQL as informações requisitadas e retorna ao sistema utilizando a notação XML.

3.2.4.1 Diagrama de classe: Gerenciamento de Menu

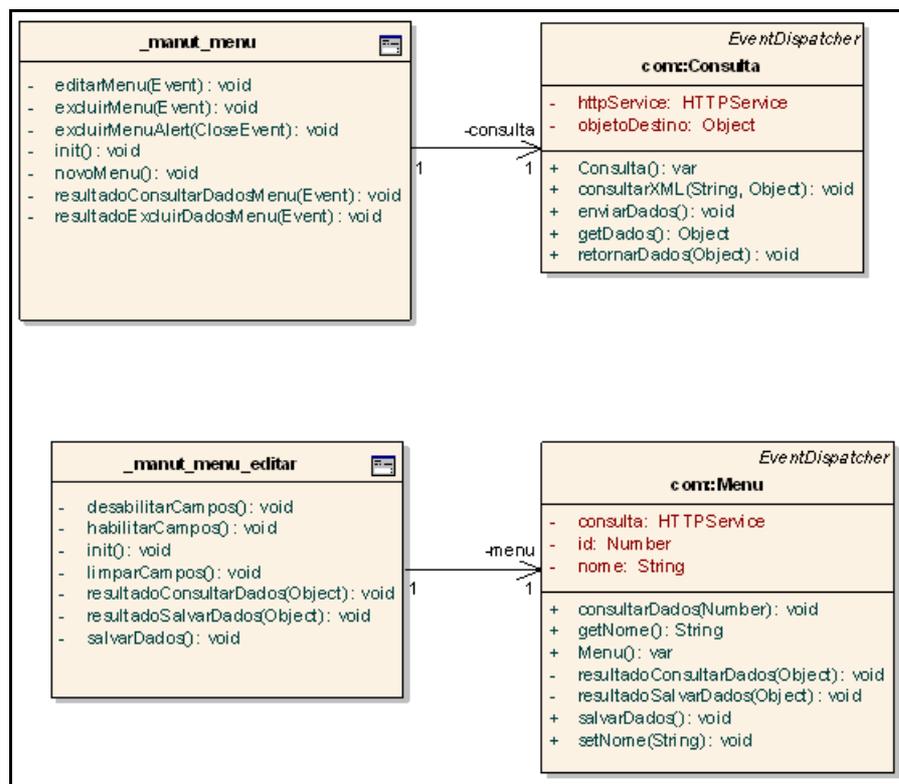


Figura 9 – Diagrama de classe de gerenciamento de menu

A Figura 9 apresenta a interface do usuário para gerenciar os menus do *website*. Inicialmente a classe que apresenta as funções de adição, edição e exclusão de itens de menu é a classe `_manut_menu`. Através dela o usuário pode executar qualquer uma destas ações logo após o recebimento do evento disparado pela classe `Consulta`, que retorna os menus já cadastrados no sistema.

Esta classe `Consulta` é uma classe utilizada para consultar as informações no banco de dados através do PHP e retornar para o usuário, no FLEX, estes dados formatados de forma legível, desta forma ela recebe um objeto de retorno em sua construtora que seria um componente utilizado para apresentar estes dados, no caso o componente `DataGrid`.

Após a escolha da opção desejada, a função `trocarTela` é executada na classe

_principal, apresentada anteriormente. Esta função recebe o nome da tela que deve ser apresentada em seguida para o usuário, no caso, a tela de edição de menus representada pela classe _manut_menu_editar. Nela são apresentados os campos de um item de menu. Caso tenha sido feito a escolha de edição de um menu já existente, a classe Menu é responsável pela consulta dos dados do menu selecionado, bem como salvar as informações alteradas ou inseridas caso a escolha tenha sido para inserção de um novo item de menu.

3.2.4.2 Diagrama de classe: Gerenciamento de Conteúdo

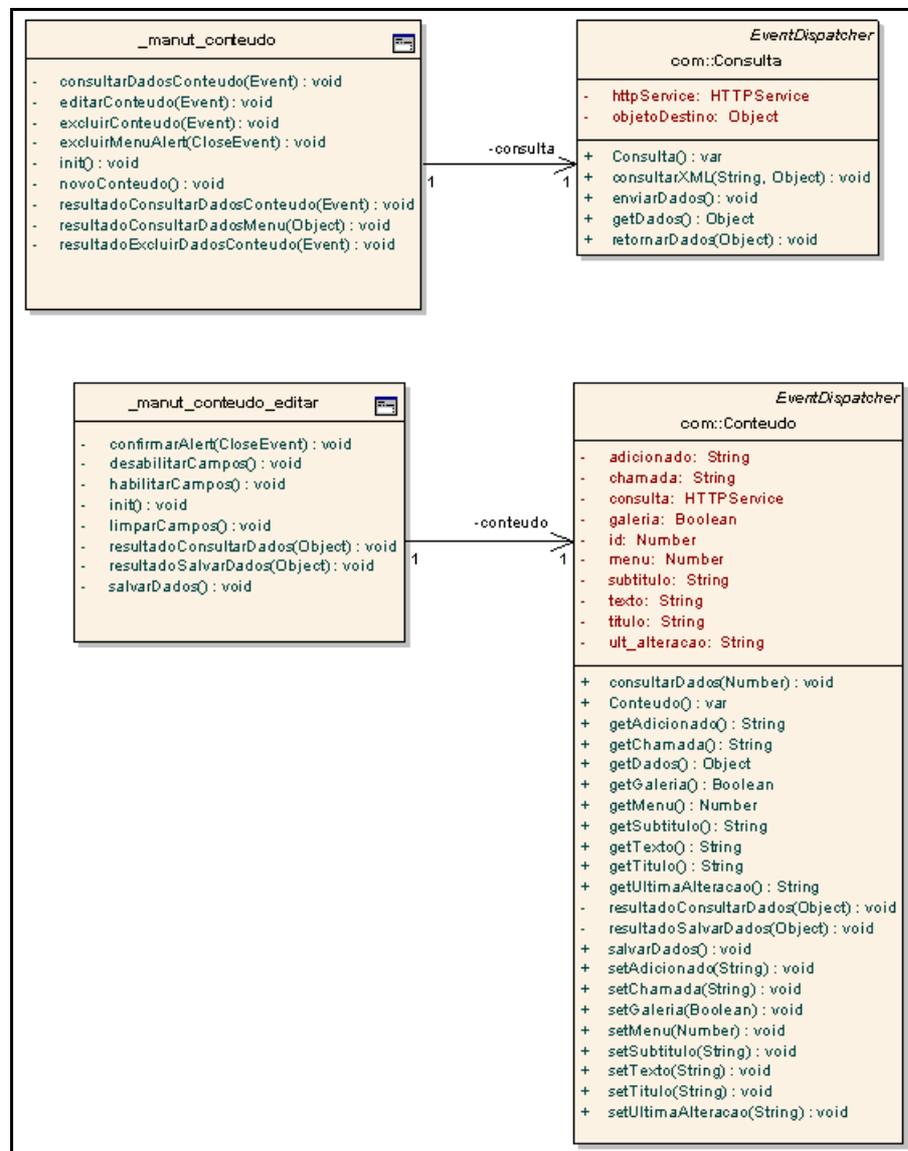


Figura 10 – Diagrama de classe de gerenciamento de conteúdo

A Figura 10 apresenta a interface do usuário para gerenciar o conteúdo de cada menu do *website*. Nela a classe *Consulta* retorna os menus cadastrados no sistema atualmente para

a classe `_manut_conteudo` e aguarda que o usuário selecione um destes menus. Logo após, a classe `Consulta` é chamada novamente para trazer os conteúdos cadastrados no menu selecionado, liberando assim as funções de adição, edição e exclusão de conteúdo.

Após a escolha da opção desejada, são apresentados os campos de conteúdo. Caso tenha sido feito a escolha de edição de um conteúdo já existente, a classe `Conteudo` é responsável pela consulta dos dados, além de salvar as informações alteradas ou inseridas caso a escolha tenha sido para inserção de um conteúdo no menu.

3.2.4.3 Diagrama de classe: Gerenciamento de Galeria de Imagens

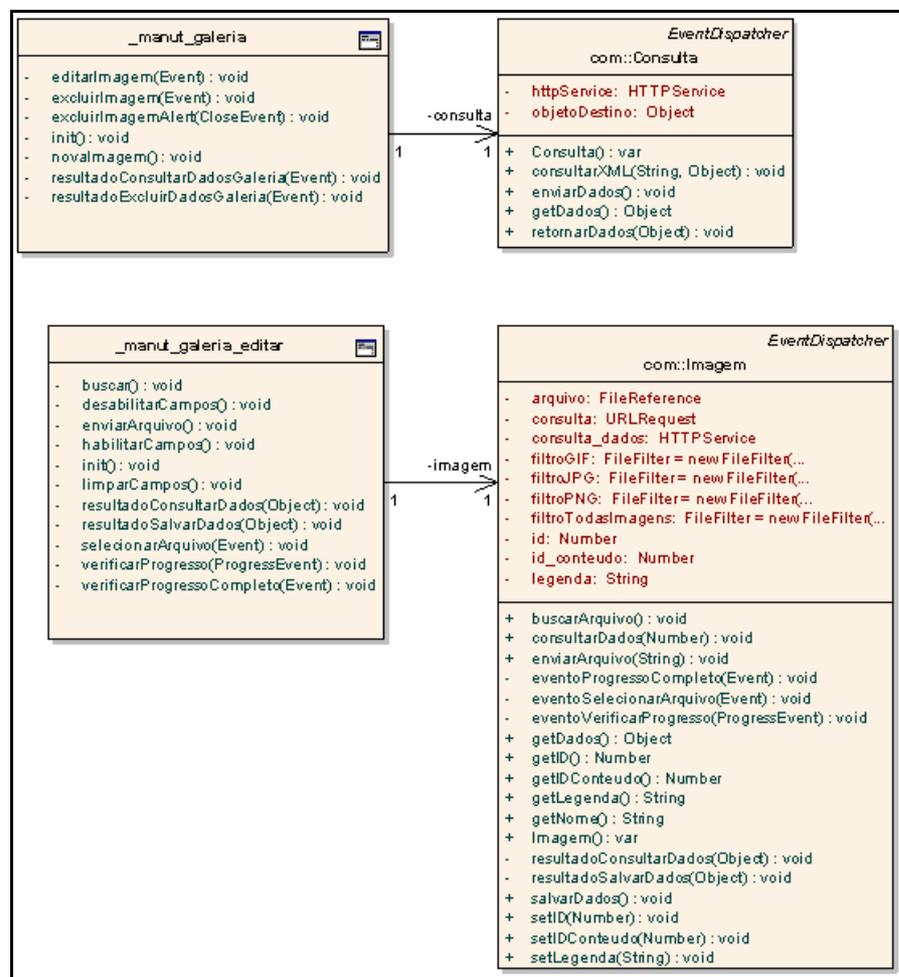


Figura 11 – Diagrama de classe de gerenciamento de galeria de imagens

A Figura 11 apresenta a interface do usuário para gerenciar uma galeria de imagem caso o usuário tenha optado por uma ao inserir um conteúdo no sistema. A classe `Consulta` retorna as imagens pertencentes à galeria do conteúdo selecionado anteriormente e libera a inserção, edição ou exclusão das imagens cadastradas no sistema.

Após a escolha da opção desejada, são apresentados os campos para inserir ou editar uma imagem. Caso tenha sido feito a escolha de edição de um conteúdo já existente, a classe *Imagem* é responsável pela consulta dos dados, além de salvar as informações alteradas e disparar os eventos de progresso de envio de um arquivo para o servidor.

3.2.4.4 Diagrama de classe: Configuração do Sistema

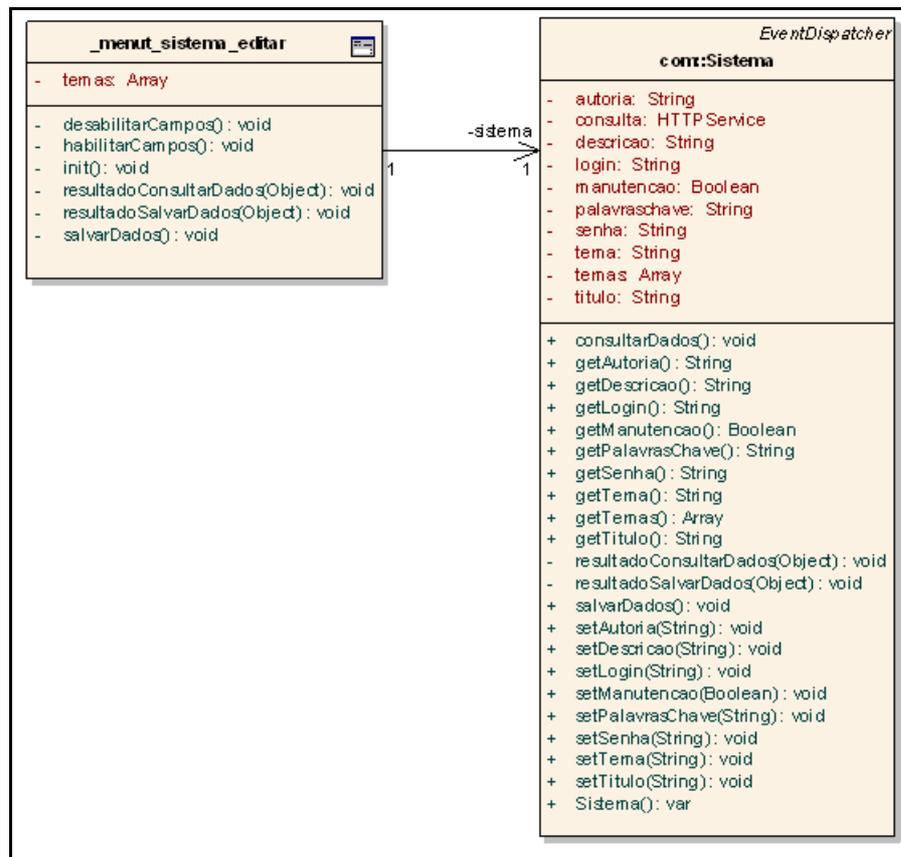


Figura 12 – Diagrama de classe de configuração do sistema

A Figura 12 apresenta a classe *_manut_sistema_editar* responsável pela interface de configuração do sistema de gerenciamento de conteúdo. Ela instancia a classe *Sistema*, responsável pela consulta destas informações de configuração e também por salva-las.

3.2.5 Diagramas de Seqüência

Os diagramas de seqüência são utilizados na UML para representar as mensagens trocadas entre as classes e/ou objetos dos mais diversos processos que ocorrem dentro de um

sistema.

A seguir são apresentados os principais diagramas de seqüência, representando os 3 principais casos de uso do sistema, adição de menu, adição de conteúdo e adição de galeria de imagens.

3.2.5.1 Diagrama de seqüência: Adição de Menu

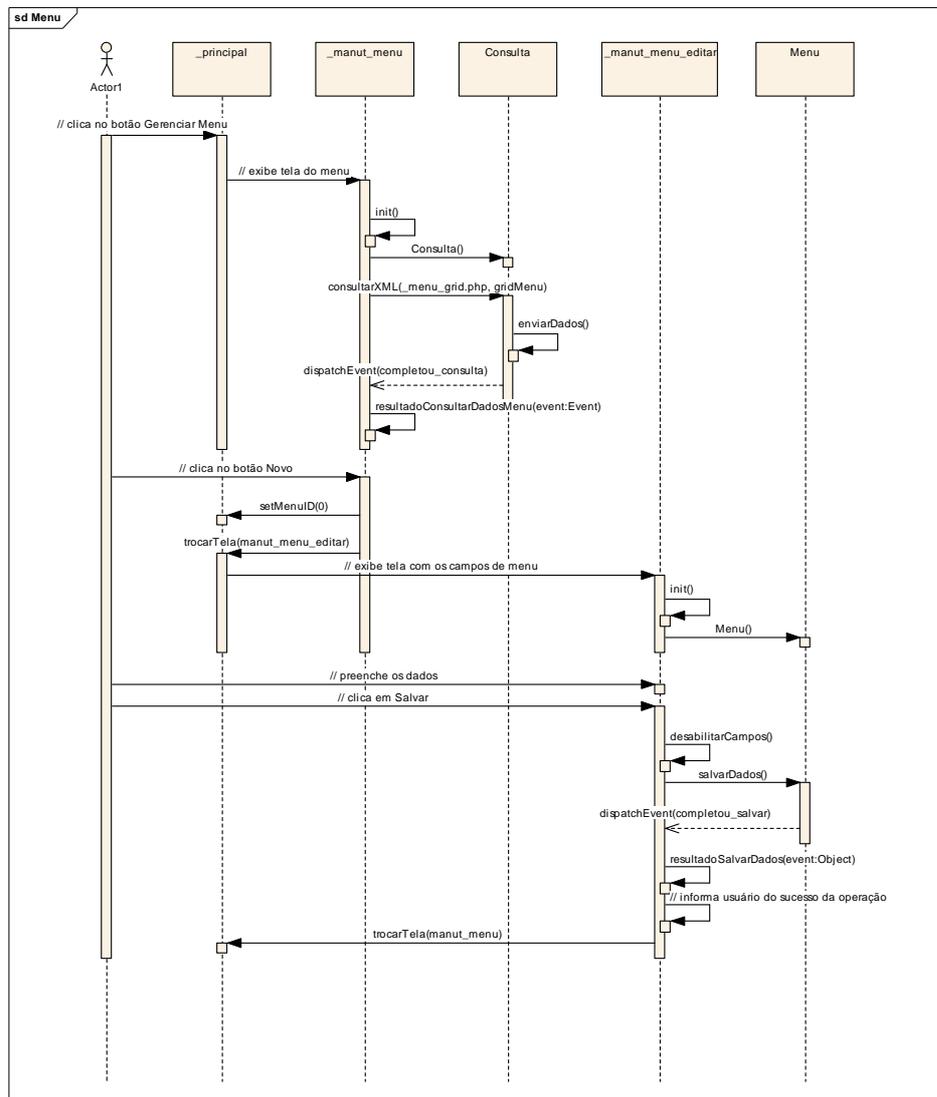


Figura 13 – Diagrama de seqüência da adição de menu

A Figura 13 mostra o fluxo das mensagens trocadas entre as classes desde o processo de seleção da opção de adicionar um menu até o preenchimento dos campos necessários para a adição de um novo menu no sistema. Este diagrama faz parte do caso de uso “UC02.02 Adicionar Menu”.

3.2.5.2 Diagrama de seqüência: Adição de Conteúdo

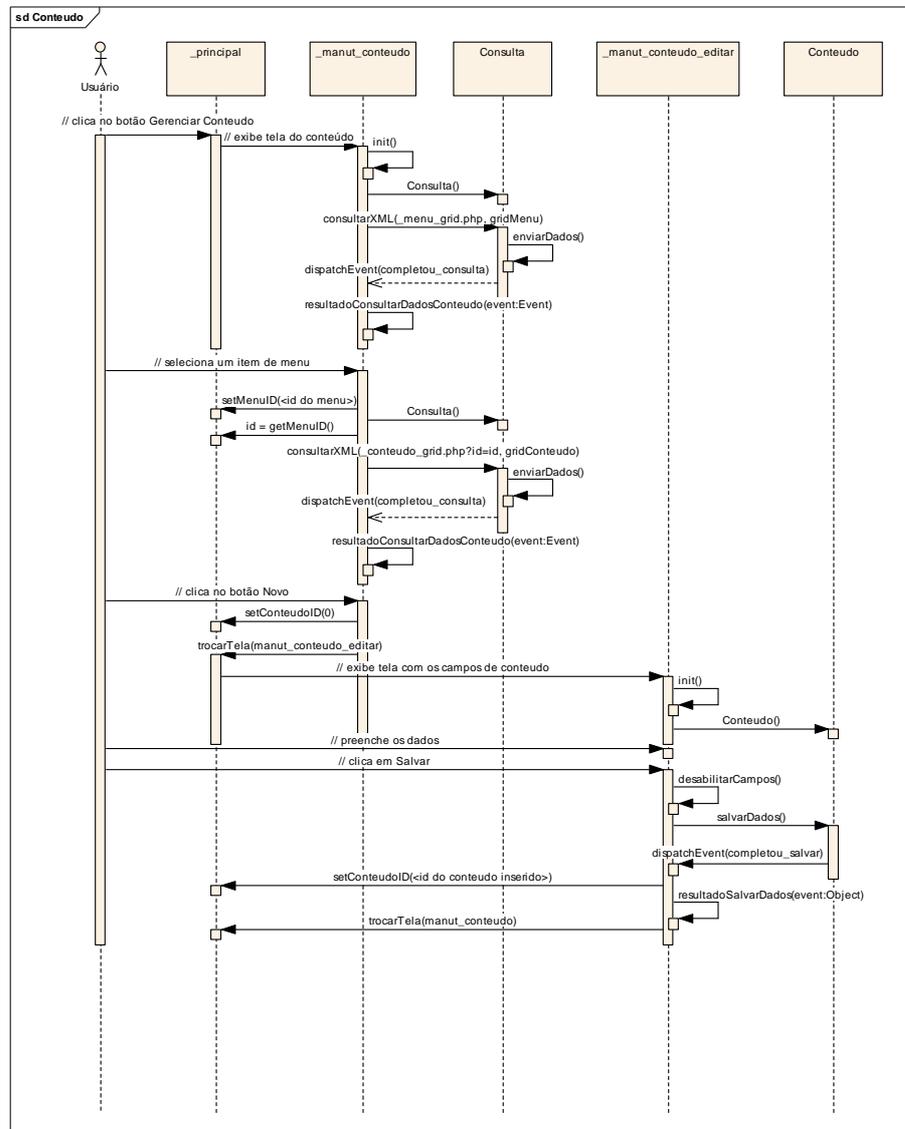


Figura 14 – Diagrama de seqüência da adição de conteúdo

Este diagrama de seqüência da Figura 14 representa a adição de um conteúdo. Da mesma forma apresentada no diagrama da Figura 13, ele mostra a troca de mensagens entre as classes desde a seleção da opção de adição de um novo conteúdo, até o término do processo no preenchimento dos campos e envio dos dados para serem salvos no banco de dados. Este diagrama faz parte do caso de uso “UC02.03 Adicionar Conteúdo”.

3.2.5.3 Diagrama de seqüência: Adição de Galeria de Imagem

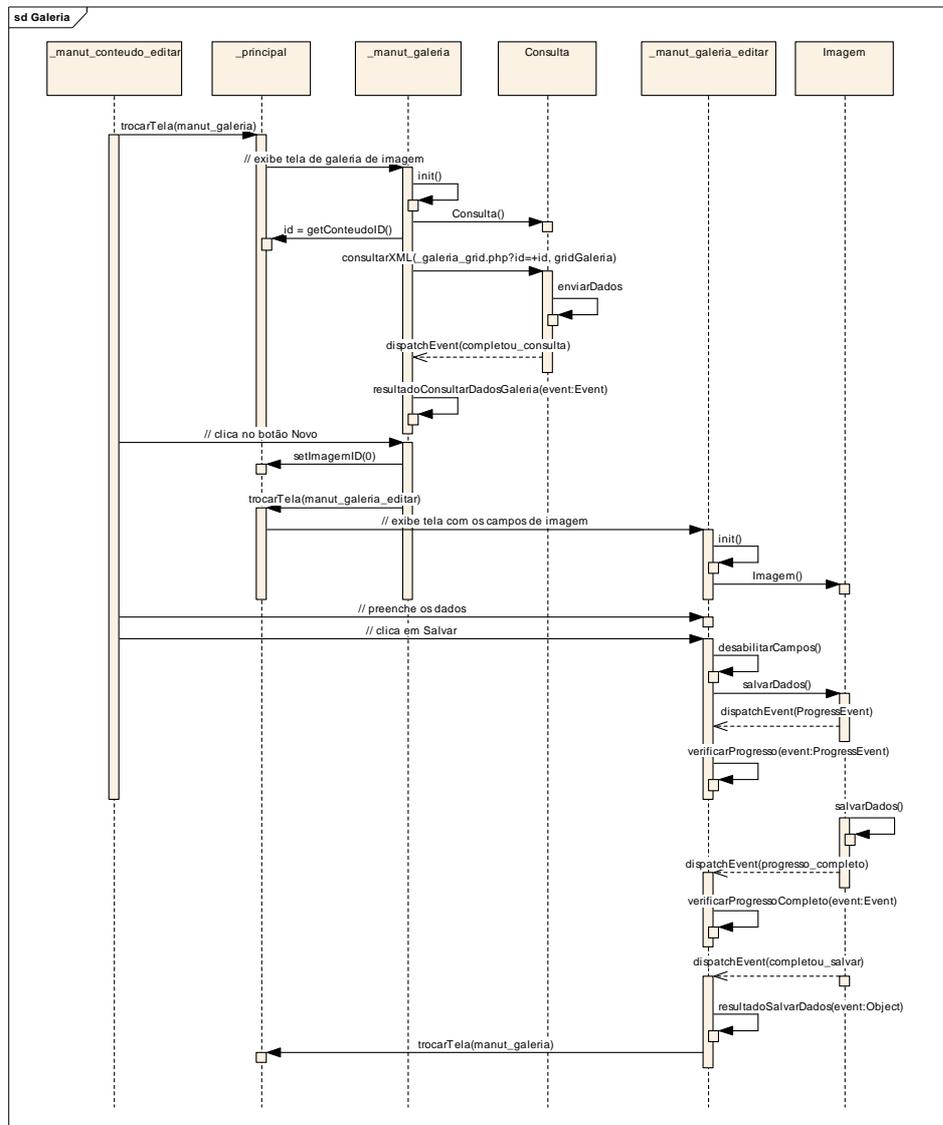


Figura 15 – Diagrama de seqüência da adição de galeria de imagem

A Figura 15 apresenta as mensagens trocadas entre as classes no processo de adição de uma galeria de imagens. A classe `_manut_conteudo_editar` é a responsável por iniciar este processo caso o usuário tenha optado por adicionar uma galeria de imagens em um determinado conteúdo. Este diagrama faz parte do caso de uso “UC02.04 Adicionar Galeria”.

3.3 IMPLEMENTAÇÃO

As seções a seguir descrevem as ferramentas utilizadas, o processo de desenvolvimento, técnicas utilizadas e o funcionamento do sistema de gerenciamento de conteúdo.

3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento do sistema foi utilizado o FLEX e algumas ferramentas auxiliares específicas que são detalhadas a seguir. Nas seções seguintes são detalhadas estas ferramentas e onde cada uma foi utilizada no processo de desenvolvimento, além da técnica utilizada para o tratamento dos *templates* de visualização do *website*.

3.3.1.1 phpMyAdmin

O phpMyAdmin (PHPMYADMIN, 2007) foi a ferramenta utilizada para monitoração das alterações ocorridas no banco de dados MySQL. Ele é um projeto de código fonte aberto implementado na linguagem de programação PHP para administrar um banco de dados MySQL na web.

Após desenvolvido o diagrama relacionado ao banco de dados utilizando o DBDesigner (FABULOUS FORCE DATABASE TOOLS, 2003), foi exportado um arquivo com os comandos SQL para a criação da base de dados. Através do phpMyAdmin foi executado estes comandos e acompanhado todas as transações feitas no MySQL. As inclusões, exclusões e alterações eram feitas pelo sistema de gerenciamento de conteúdo e o phpMyAdmin foi utilizado para confirmar estas alterações, corrigir eventuais erros de consulta no banco de dados e dar manutenção a base de dados quando necessário.

3.3.1.2 FLEX

O FLEX é o ambiente de programação do sistema de gerenciamento de conteúdo proposto. Ele foi desenvolvido pela Adobe (ADOBE SYSTEMS INCORPORATED, 2007) baseado no ambiente de programação conhecido por Eclipse (ECLIPSE, 2007), desta forma o FLEX herda todas as características do mesmo. Sendo assim, ele disponibiliza um ambiente para desenvolvimento que se utiliza de *plugins* e permite a extensão do escopo original do ambiente para diversos outros propósitos.

Nele foi desenvolvida toda a aplicação do gerenciamento de conteúdo, desde as classes e objetos utilizados, além de toda a interface de interação com o usuário. A comunicação com o PHP também foi feita utilizando requisições HTTP, permitindo a integração do sistema com o banco de dados MySQL.

3.3.1.3 Dreamweaver

O Dreamweaver também é uma ferramenta desenvolvida pela Adobe, que tem como propósito o desenvolvimento de *websites*. Ele permite criar arquivos HTML e programar utilizando as linguagens ASP, ASP.NET, PHP, ColdFusion e JavaScript.

A escolha do Dreamweaver se deu pela facilidade em programar utilizando a linguagem PHP e por permitir visualizar, dentro da ferramenta, como está sendo renderizado o código fonte. Desta forma as criações dos *templates* para o *website* foram mais simples e rápidos do que utilizando uma ferramenta sem este recurso.

Os arquivos exemplos de XML também foram criados nesta ferramenta para testes no sistema em FLEX. Posteriormente estes arquivos foram substituídos por arquivos programados em PHP que geravam os arquivos XML necessários para o sistema consultando as informações do banco de dados. A Figura 16 mostra um destes arquivos fazendo a consulta ao banco de dados e montando o arquivo XML para o sistema.

```

1 <?php
2 require_once("../config.php");
3
4 $id = $_GET['id'];
5
6 $consulta = mysql_query("SELECT * FROM conteudo WHERE idconteudo = '{$id}'");
7 if (mysql_num_rows($consulta) > 0)
8 {
9     print('<?xml version="1.0" encoding="utf-8"?>');
10    print('<registros>');
11    while ($row = mysql_fetch_assoc($consulta))
12    {
13        print('<registro>');
14        print('    <idmenu> ' . utf8_encode($row['menu_idmenu']) . '</idmenu>');
15        print('    <titulo> ' . utf8_encode($row['titulo']) . '</titulo>');
16        print('    <subtitulo> ' . utf8_encode($row['subtitulo']) . '</subtitulo>');
17        print('    <chamada> ' . utf8_encode($row['chamada']) . '</chamada>');
18        print('    <texto><![CDATA[ ' . utf8_encode($row['texto']) . ' ]]></texto>');
19        print('    <galeria> ' . utf8_encode($row['galeria']) . '</galeria>');
20        print('    <adicionado> ' . utf8_encode($row['adicionado']) . '</adicionado>');
21        print('    <ult_alteracao> ' . utf8_encode($row['ult_alteracao']) . '</ult_alteracao>');
22        print('</registro>');
23    }
24    print('</registros>');
25 }
26 ?>

```

Figura 16 – Arquivo XML sendo montado utilizando PHP

Outra observação a ser feita sobre esta montagem dos arquivos XML é o uso das funções nativas do PHP `utf8_encode` e `utf8_decode`. Estas funções são responsáveis por codificar e decodificar informações de textos e/ou caracteres no formato *8-bit UCS/Unicode Transformation Format* (UTF-8).

O UTF-8 é um padrão de codificação de caracteres que permite representar qualquer caractere universal no padrão Unicode. É um dos formatos mais comuns em e-mails, *websites* e outros lugares em que são guardados dados de texto.

No FLEX não é diferente, o formato padrão assumido por ele é UTF-8. Este formato não é o único suportado pelo FLEX, mas é o padrão. Desta forma optou-se por manter este padrão e fazer esta codificação dos dados, ou decodificação.

3.3.1.4 ErgoList

O ErgoList, como mencionado anteriormente, é um *checklist* para a avaliação de interfaces homem-computador. Desta forma o ErgoList foi utilizado para a verificação da interface criada e correção de eventuais detalhes que poderiam dificultar o uso do sistema.

A interface projetada para o trabalho foi feita da forma mais simples e direta possível, pensando na facilidade de uso e evitando menus complicados ou com inúmeras opções, poluindo visualmente e impossibilitando uma boa leitura. O ErgoList foi responsável por

direcionar estas idéias na elaboração da interface do sistema com o usuário.

3.3.1.5 Templates

Os *templates* são utilizados para determinar a aparência de um *website*. Ele demarca e separa onde ficam as áreas de apresentação do conteúdo e as áreas referentes a aparência do site com imagens, tabelas, fotos e estilos.

Na estrutura de pastas do *website*, existe a pasta “temas”, que possui outros diretórios. Cada diretório dentro desta pasta é considerado um *template* diferente. O nome da pasta serve como nome para o *template*, que é oferecido para ser alterado na tela de configuração do sistema no gerenciador de conteúdo.

Caso o usuário queira alterar algum destes *templates*, basta copiar um dos existentes para uma outra pasta com outro nome, e editar o arquivo index.html em qualquer editor de HTML que desejar. O sistema não faz restrição alguma com relação a imagens, figuras ou forma de distribuição de conteúdo, a única restrição feita são as palavras chaves utilizadas para indicar a posição de cada conteúdo consultado quando for necessária a visualização do *website*.

Estes *templates* possuem palavras reservadas que são utilizadas para serem substituídas pelo conteúdo do *website*. A Figura 17 mostra o código fonte sem a substituição das palavras reservadas.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5 <title>
6
7 [titulo]
8
9 </title>
10 <style ...
76 <td></td>
77 <td width="525"></td>
78 </tr>
79 </table>

```

Figura 17 – Arquivo de *template* sem a substituição das palavras reservadas

As palavras reservadas [titulo] e [pasta_template] estão no código fonte do template para serem substituídas pelas informações cadastradas no sistema de gerenciamento de conteúdo. A Figura 18 mostra como fica o código fonte depois da substituição destas palavras reservadas.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5 <title>
6
7 TITULO DO SITE
8
9 </title>
10 <style ...
76 <td></td>
77 <td width="525"></td>
78 </tr>
79 </table>

```

Figura 18 – Arquivo de *template* com a substituição das palavras reservadas

As duas palavras reservadas nas linhas 7, 76 e 77 foram substituídas pelos seus respectivos conteúdos gravados no banco de dados.

3.3.2 Operacionalidade da implementação

A operacionalidade da implementação é demonstrada através da utilização do sistema de gerenciamento de conteúdo desde a configuração dentro do sistema, passando pela adição de um menu, um conteúdo para este menu e uma galeria de imagens para o mesmo.

Quando a página inicial do sistema de gerenciamento de conteúdo é aberta, o usuário deve aguardar o sistema carregar. Este processo ocorre apenas uma vez em toda a utilização do sistema e em nenhum outro momento será necessário carregar a interface ou o sistema novamente, exceto quando a página do sistema for fechada e acessada novamente.

Uma vez carregado o sistema, a tela de autenticação do usuário é apresentada conforme a Figura 19. Nela o usuário deve fornecer o *login* e a senha para acesso as funções de manutenção do sistema de gerenciamento de conteúdo. Caso esta seja a primeira vez que o sistema está sendo executado, o usuário utiliza o *login* e senha padrões, que podem ser alterados nas configurações do sistema.

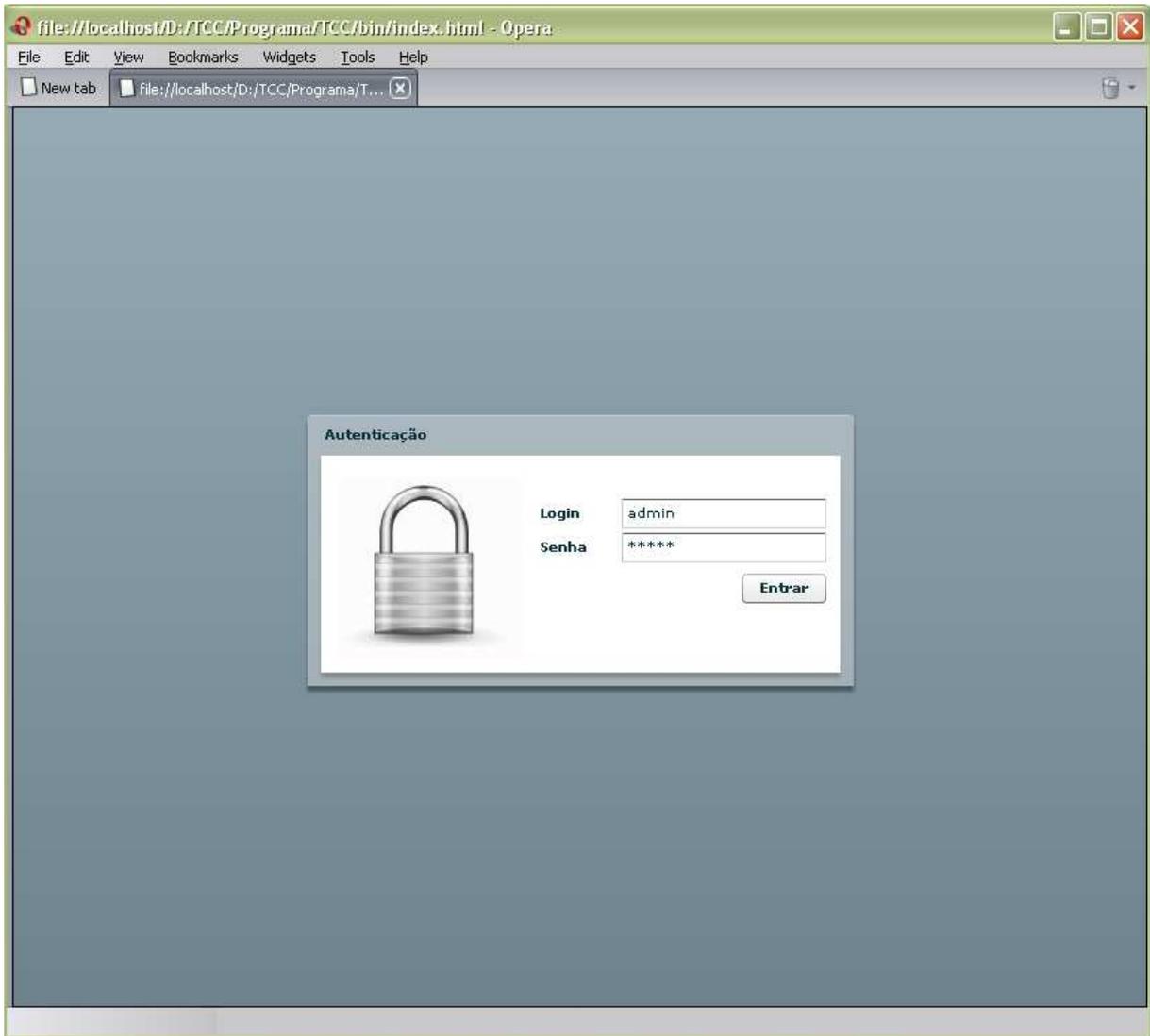


Figura 19 – Autenticação do usuário

Caso a autenticação do usuário tenha acontecido com sucesso, o usuário é levado à tela principal do sistema que possui acesso as funcionalidades do mesmo. A Figura 20 mostra esta tela com os botões de acesso as funções de configuração do sistema, gerenciamento de menu e gerenciamento de conteúdo.

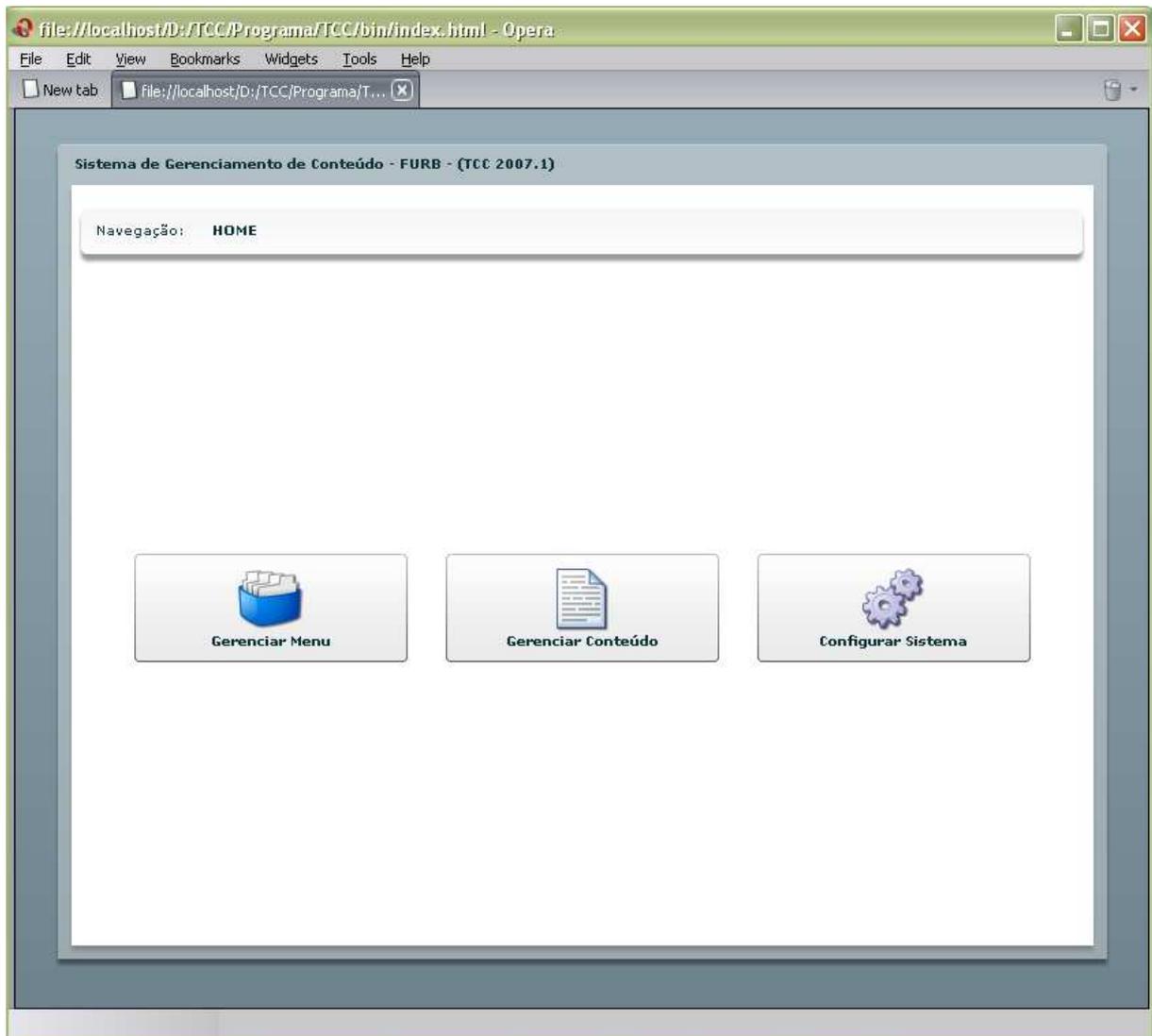


Figura 20 – Tela principal do sistema

O primeiro passo que o usuário deve tomar é acessar o item “Configurar Sistema” para alterar as preferências do *website*. Nela são apresentadas as configurações atuais do sistema de gerenciamento de conteúdo, como pode ser observado na Figura 21. O usuário tem a possibilidade de alterar o login e senha de acesso, as informações de indexação dos sistemas de buscas, título do *website*, texto para direitos autorais e o *template* de visualização do *website*.

file://localhost/D:/TCC/Programa/TCC/bin/index.html - Opera

File Edit View Bookmarks Widgets Tools Help

New tab file://localhost/D:/TCC/Programa/T...

Sistema de Gerenciamento de Conteúdo - FURB - (TCC 2007.1)

Navegação: HOME

Configurar Sistema:

Informe e/ou altere os dados do sistema nos campos abaixo:

Login: admin

Senha: *****

Título do website: Teste de apresentação

Descrição para sistemas de busca: Site que possui gerenciador de conteúdo feito em flex. Demonstração de uso da ferramenta e integração com PHP e MySQL.

Palavras chave para sistemas de busca: cms, conteúdo, flex, sistema, php, mysql

Obs.: Utilize "," para separar as palavras chave.

Autoria (direitos autorais no rodapé): Todos os direitos reservados...

Website em manutenção? Sim

Temas: template01 (selected), template01, template02

Voltar Salvar

Figura 21 – Tela de configuração do sistema

A opção para colocar o *website* em manutenção impede que qualquer pessoa veja o site apresentando uma frase informando que o *website* em questão está em manutenção e deverá voltar em breve. Esta função é utilizada principalmente quando está havendo muitas alterações de menu ou conteúdo no site e o administrador e/ou usuário do sistema não deseja que as pessoas vejam conteúdo incorretos no ar.

Depois de configurado o sistema, o usuário pode voltar para o menu inicial do sistema clicando em voltar, ou salvar as informações clicando em “Salvar”. Caso ele opte por salvar, o sistema altera o cursor do mouse para um relógio e bloqueia os campos, indicando que ele está ocupado fazendo uma transação com o banco de dados, neste caso, salvando as informações de configuração. Após o término, o sistema libera novamente os campos, retorna

o cursor do mouse ao ponteiro padrão e informa ao usuário o sucesso da operação esmaecendo o fundo e dando foco na janela de aviso, conforme pode ser visto na Figura 22.

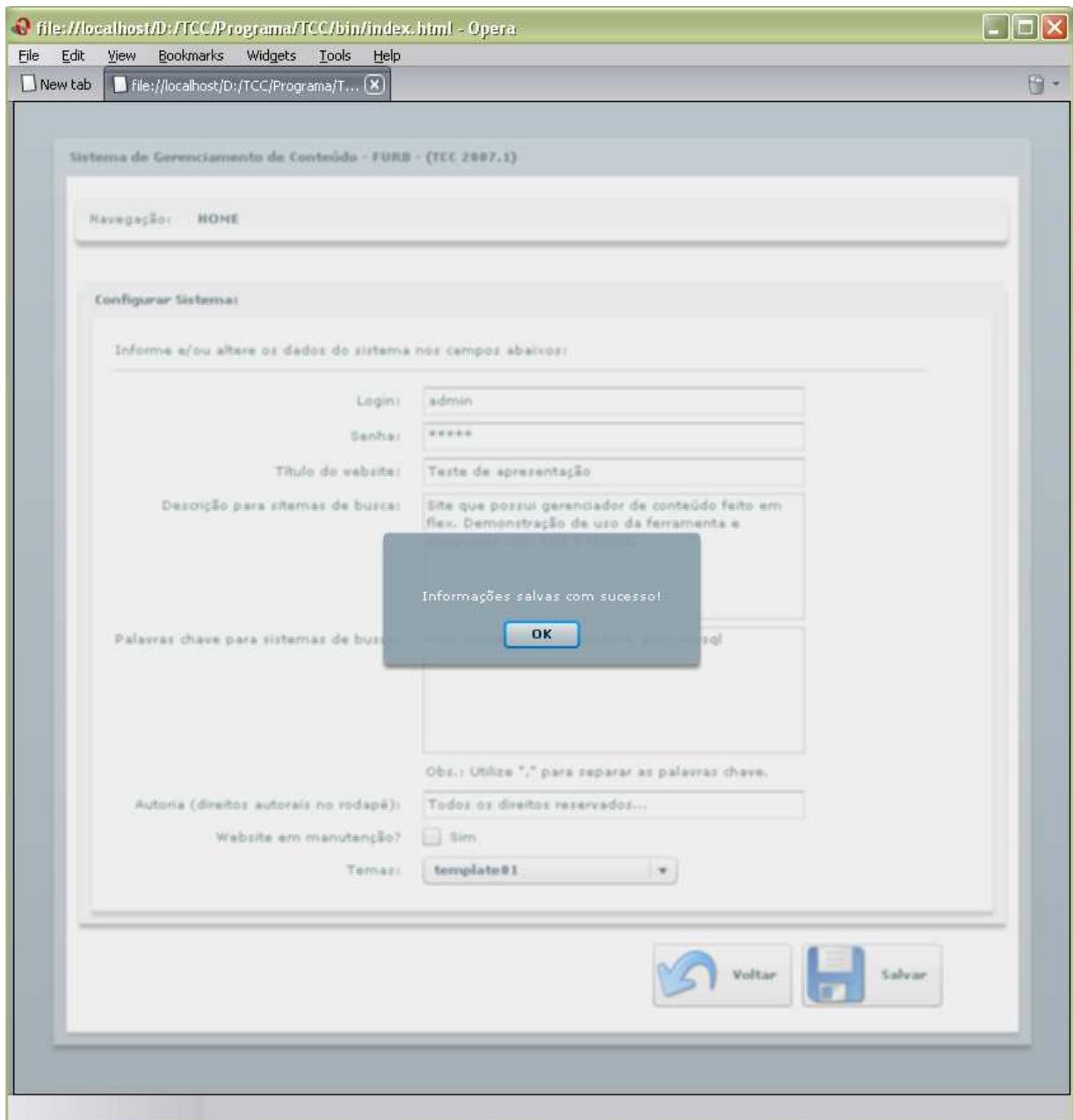


Figura 22 – Sistema confirmando sucesso da operação

Esta forma de confirmação é padrão para todo o sistema sempre que o usuário desejar salvar alguma configuração ou enviar novas informações. Para sair desta tela, o usuário clica no botão “OK” e depois no botão “Voltar” para retornar a tela principal do sistema, apresentado anteriormente na Figura 20.

Para inserir um novo menu no sistema, o usuário deve clicar no botão “Gerenciar Menu”, desta forma o sistema irá apresentar a tela de gerenciamento de menu do sistema. A primeira ação tomada pelo sistema é consultar os itens de menu existentes no banco de dados e apresentá-los ao usuário. Assim que terminar a consulta o sistema apresenta os menus

encontrados, conforme pode ser visto na Figura 23.

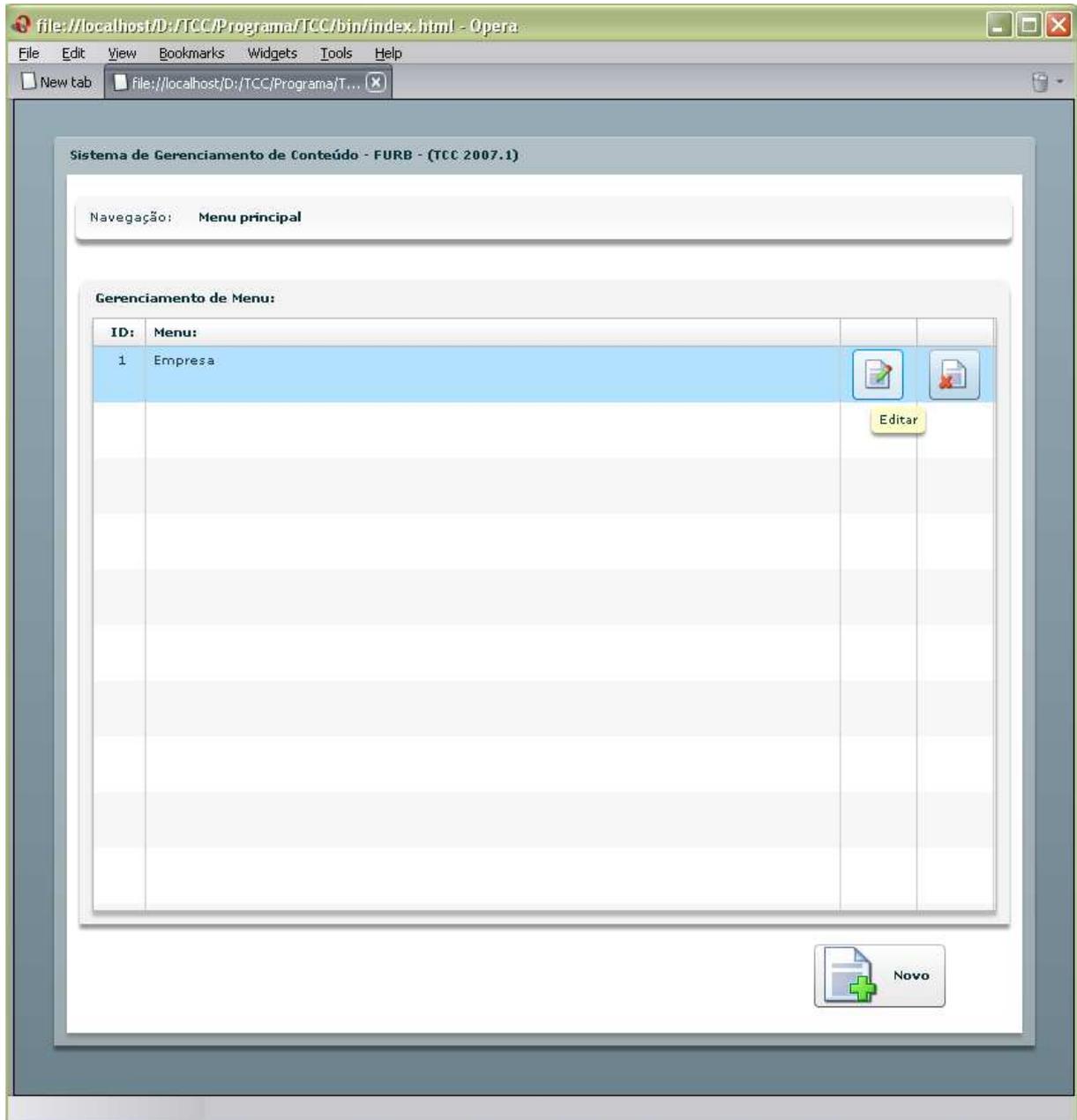


Figura 23 – Gerenciamento de menu

Em nosso caso de uso, já existe um menu no banco de dados que atende pelo nome de “Empresa”. O usuário pode editar ou excluir este menu, ou criar um novo item clicando no botão “Novo”. Ao fazer isto o sistema apresenta a tela com os campos de um item de menu conforme Figura 24.

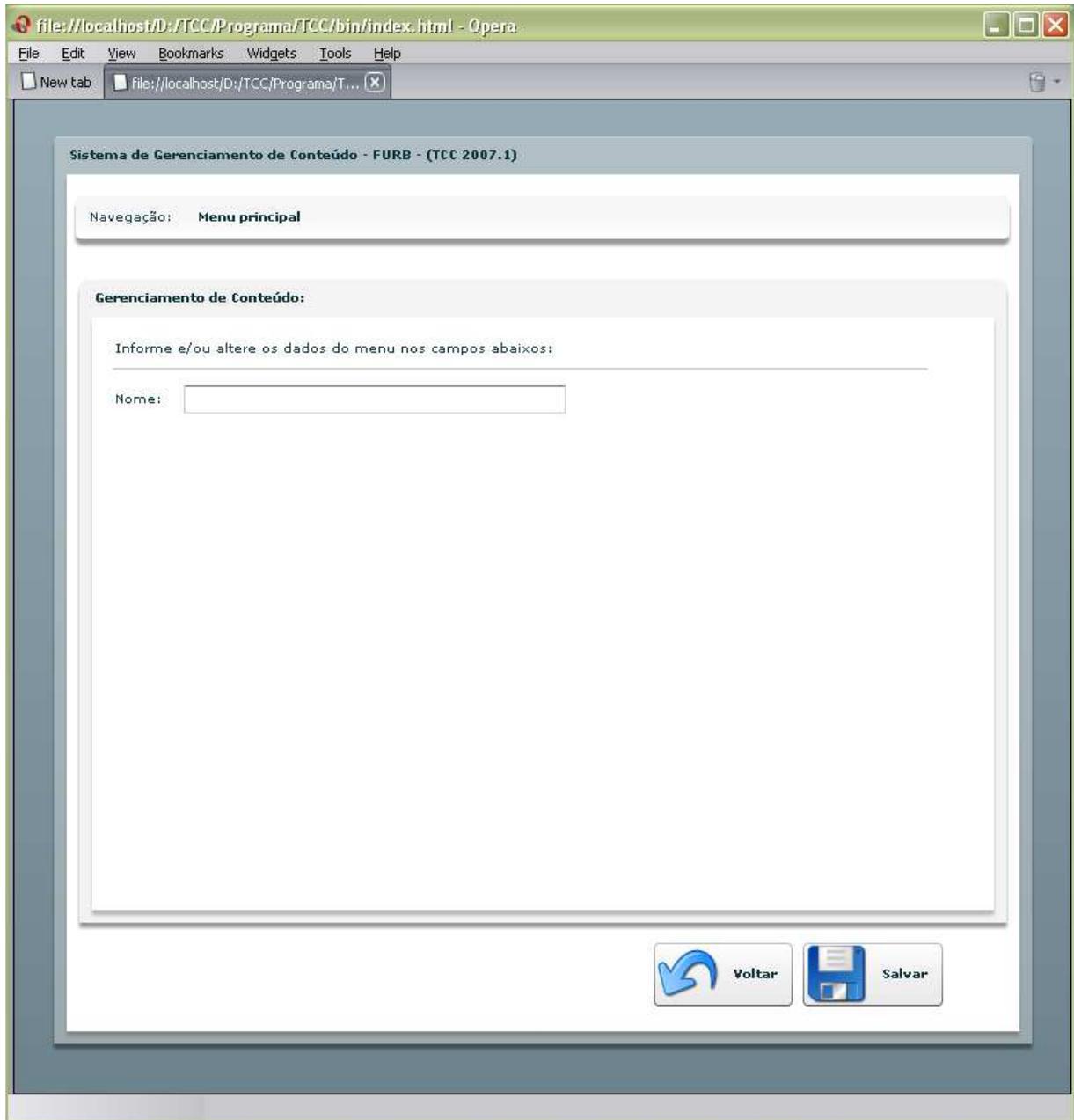


Figura 24 – Inserção de um novo item de menu

O cadastro de um novo item de menu é bem simples e requer apenas o nome do menu. Após o envio o sistema informa o sucesso da operação e retorna para o gerenciamento de menu com o novo item cadastrado.

Para o processo de adição de conteúdo o usuário deve retornar ao menu principal clicando no botão “Menu Principal” que se localiza no topo da aplicação e clicar no botão Gerenciamento de Conteúdo. O sistema irá apresentar a tela de gerenciamento de conteúdo consultando os menus cadastrados no sistema e irá aguardar a interação do usuário para que ele faça a escolha do menu desejado. Quando esta seleção ocorrer, o menu selecionado ficará em destaque conforme Figura 25, o sistema então irá consultar os conteúdos pertencentes ao item de menu selecionado, caso não encontre nenhum ele apenas informa ao usuário que não

há conteúdo cadastrado neste menu ainda.

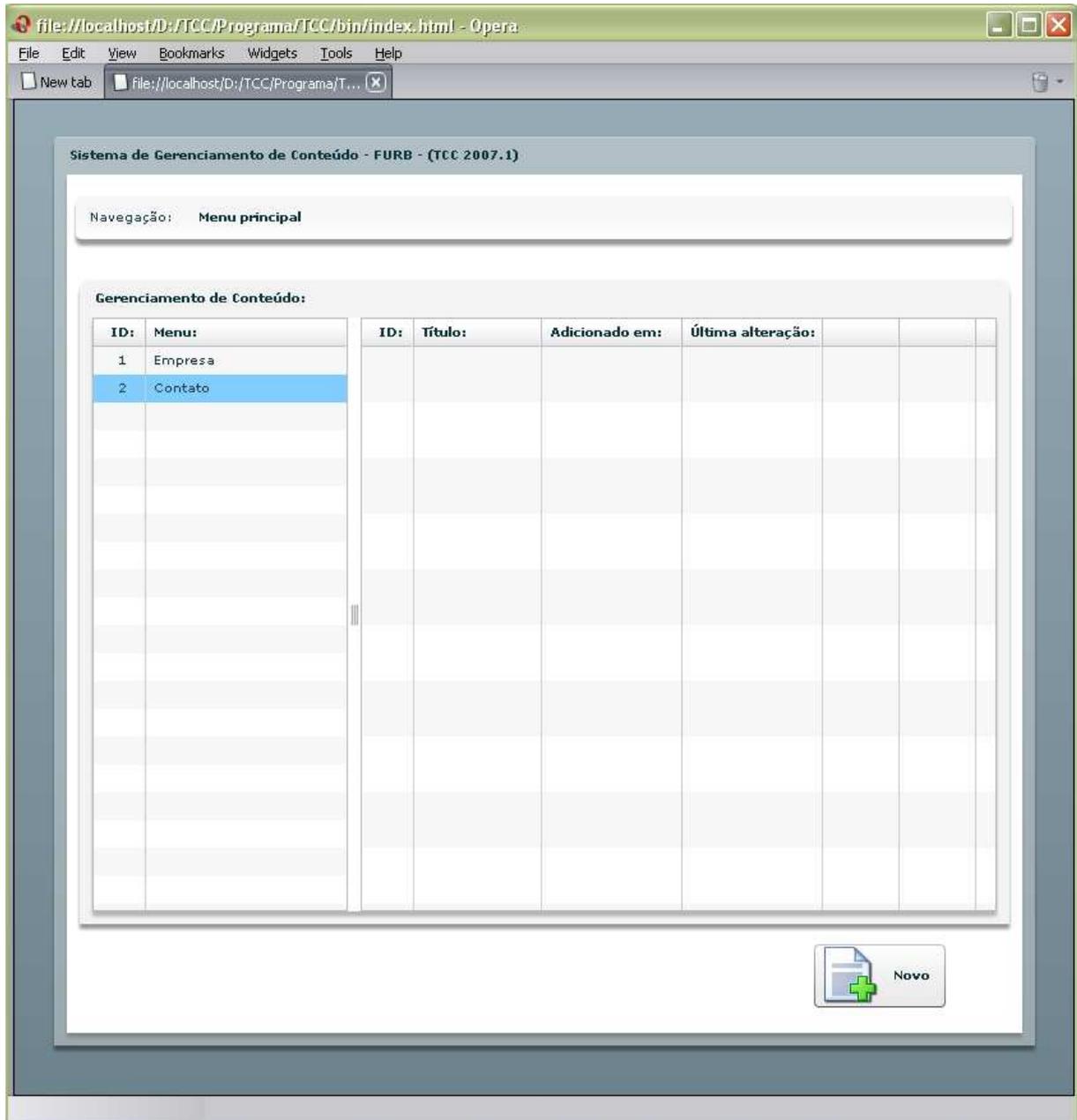


Figura 25 – Gerenciamento de conteúdo com um item de menu selecionado

Como o menu selecionado não apresenta nenhum conteúdo cadastrado, o sistema não apresenta nada no *grid* da direita. Desta forma, ou o usuário seleciona outro menu, ou clica em “Novo” para adicionar um conteúdo. Quando o fizer, o sistema irá apresentar a tela com os campos de inserção de conteúdo, conforme pode ser visto na Figura 26.

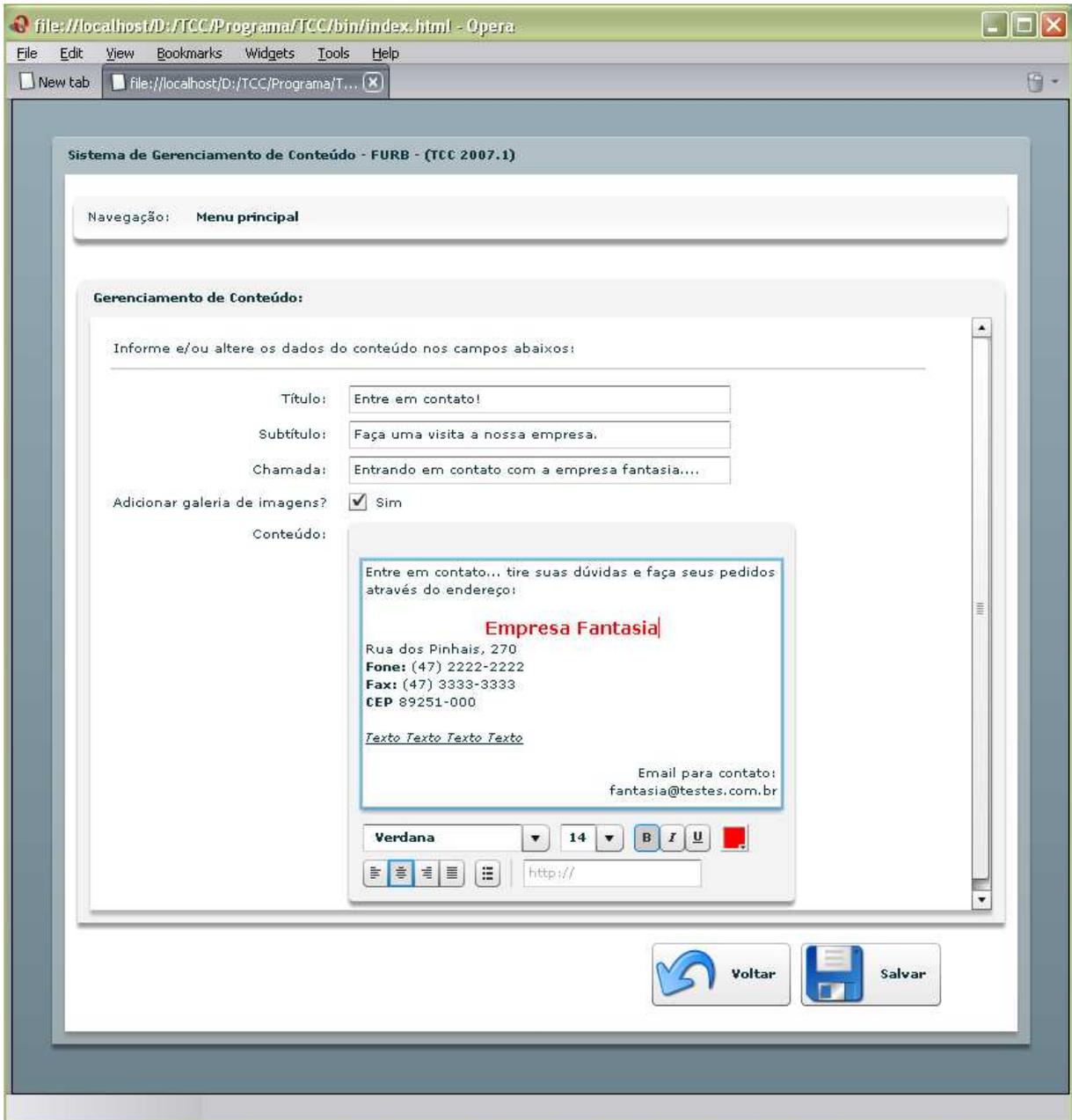


Figura 26 – Inserção de um novo conteúdo

O cadastro de um conteúdo possui vários campos para o usuário preencher. Título, subtítulo, Chamada são campos de texto normais. O item Adicionar galeria de imagens deve ser selecionado caso o usuário deseje adicionar uma galeria de imagens, caso não queira a opção deve ficar desmarcada. O campo de Conteúdo permite a edição do texto desejado com mais facilidade em termos de formatação, permitindo negritar o texto, mudar a fonte, centralizar, mudar a cor e inserir *links* para outros lugares caso seja de interesse do usuário. Da mesma forma que no cadastro de um item de menu, o usuário pode cancelar saindo desta tela clicando em “Voltar” ou “Salvar” para salvar as informações preenchidas nos campos.

Após o envio dos dados, o sistema informa que tudo ocorreu normalmente e verifica se o usuário desejou uma galeria de imagens. Em caso positivo, o usuário é encaminhado para a

tela de gerenciamento de imagens conforme Figura 27.

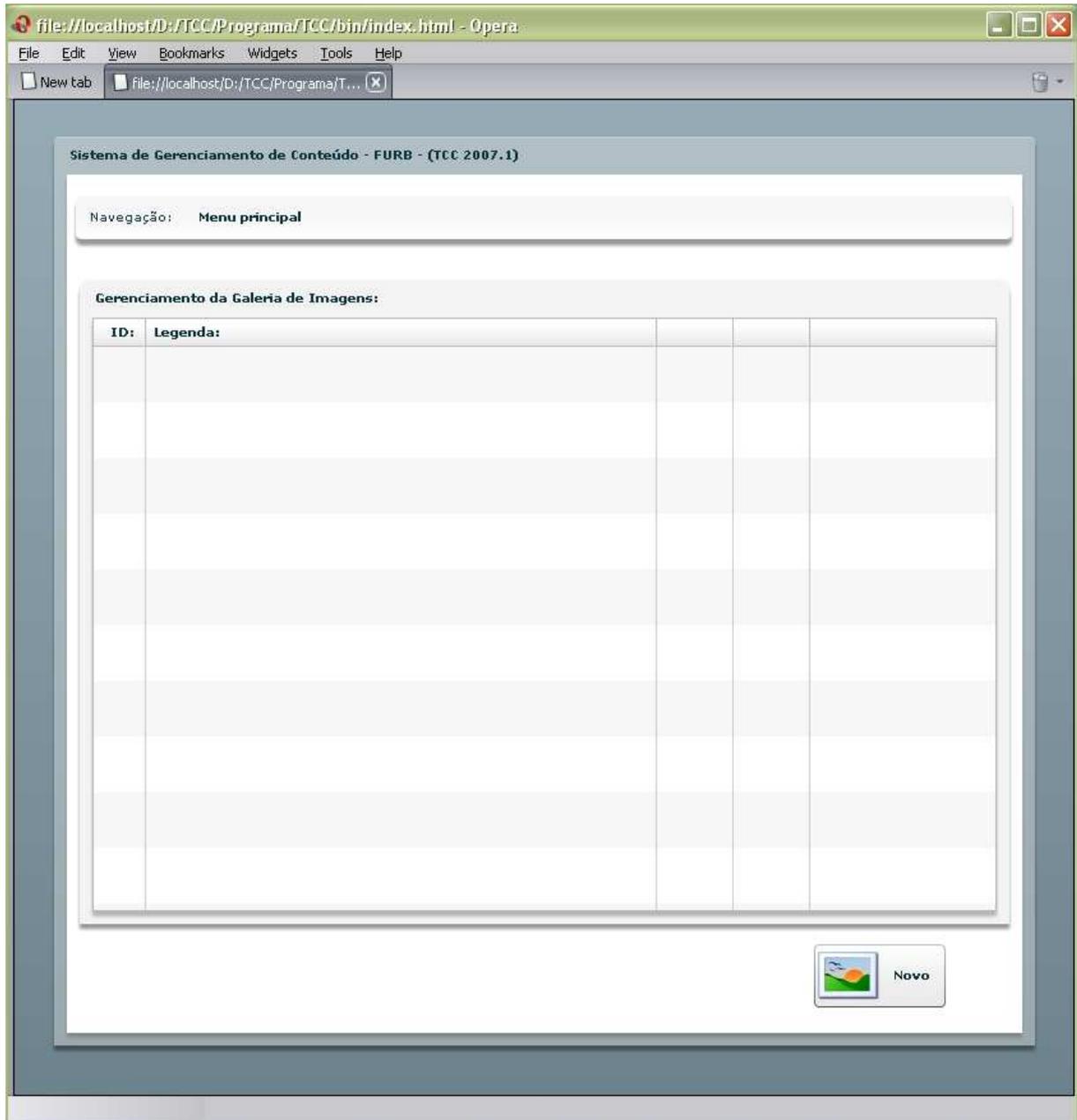


Figura 27 – Gerenciamento da Galeria de Imagens

O usuário na primeira vez que ele insere uma galeria de imagens, ele se depara com um aviso do sistema informando que não foi encontrada nenhuma imagem nesta galeria e o *grid* com as imagens cadastradas e suas respectivas legendas fica vazio. Ao clicar em “Novo” para inserir um novo retrato, ou foto, ou imagem, o usuário é encaminhado para a tela de inserção de imagem conforme Figura 28.

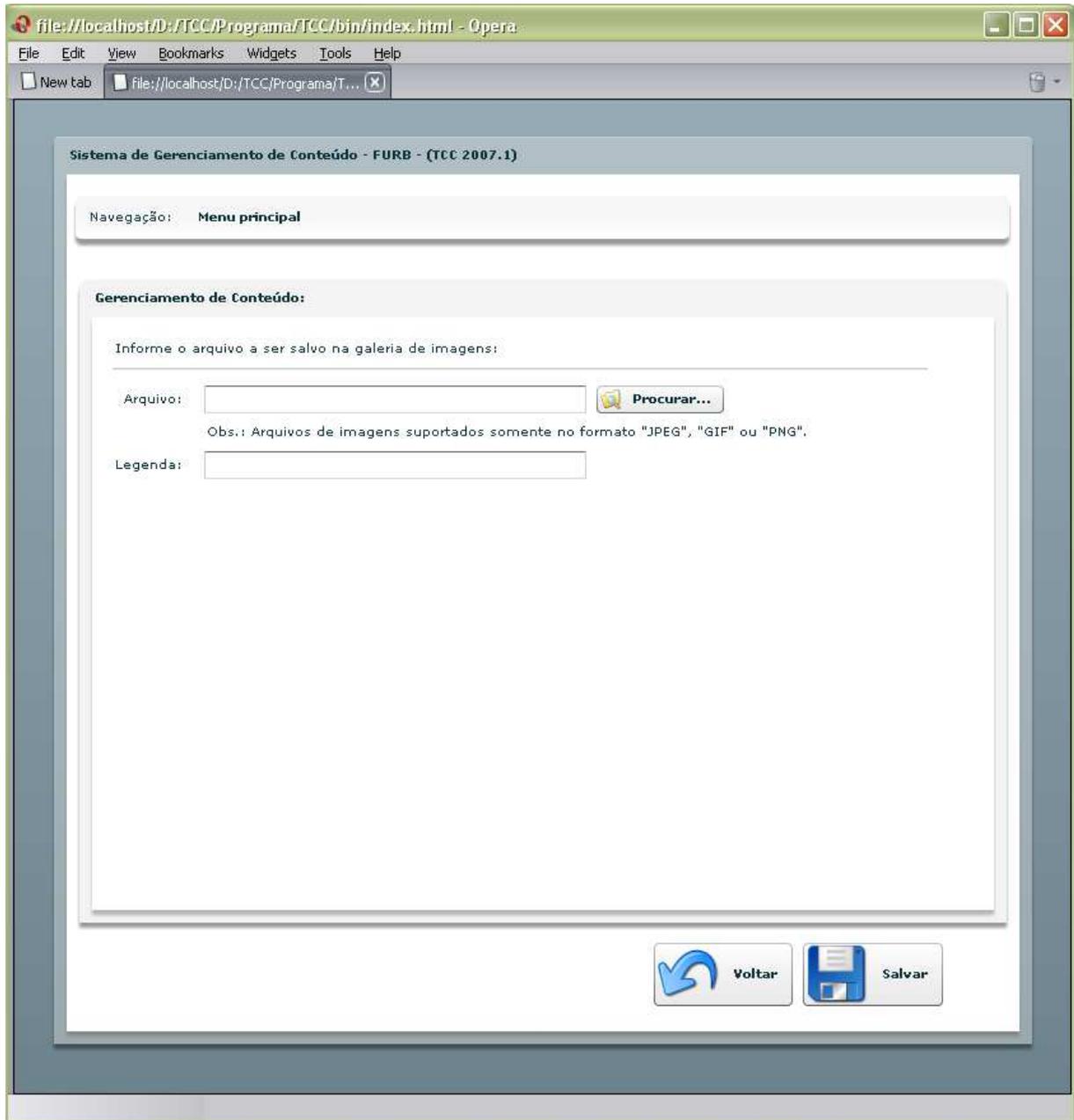


Figura 28 – Inserção de uma imagem na galeria

São poucos campos para preencher na inserção de uma foto na galeria. Basta o usuário preencher a legenda da foto que vai ser enviada para o servidor e clicar em “Procurar...” para escolher a foto desejada em seu computador. O sistema automaticamente limita o número de opções que o usuário pode escolher ao enviar as fotos, permitindo apenas os tipos de arquivos suportados como pode ser visto na Figura 29.

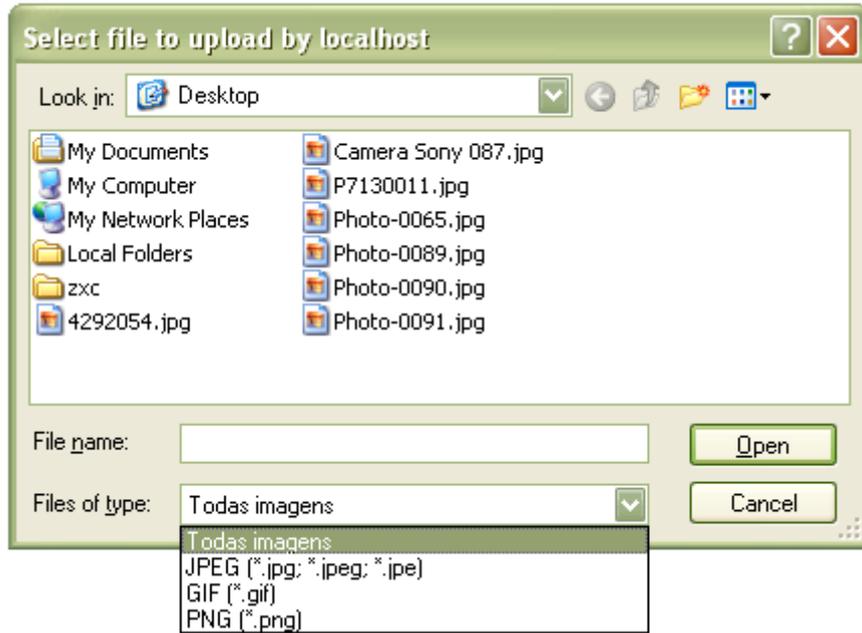


Figura 29 – Escolha do arquivo a ser enviado

Após a escolha do arquivo ter sido feita, o sistema retorna para a tela de envio da imagem com o nome do arquivo preenchido. O usuário pode modificar qualquer informação antes de enviar ao sistema esta imagem. Quando desejar, basta clicar em “Salvar” que o sistema irá salvar as informações no banco de dados e começar a enviar a imagem para o sistema. Um grande diferencial do sistema aparece neste caso, onde ele apresenta uma barra de progresso do envio da imagem, apresentado na Figura 30.

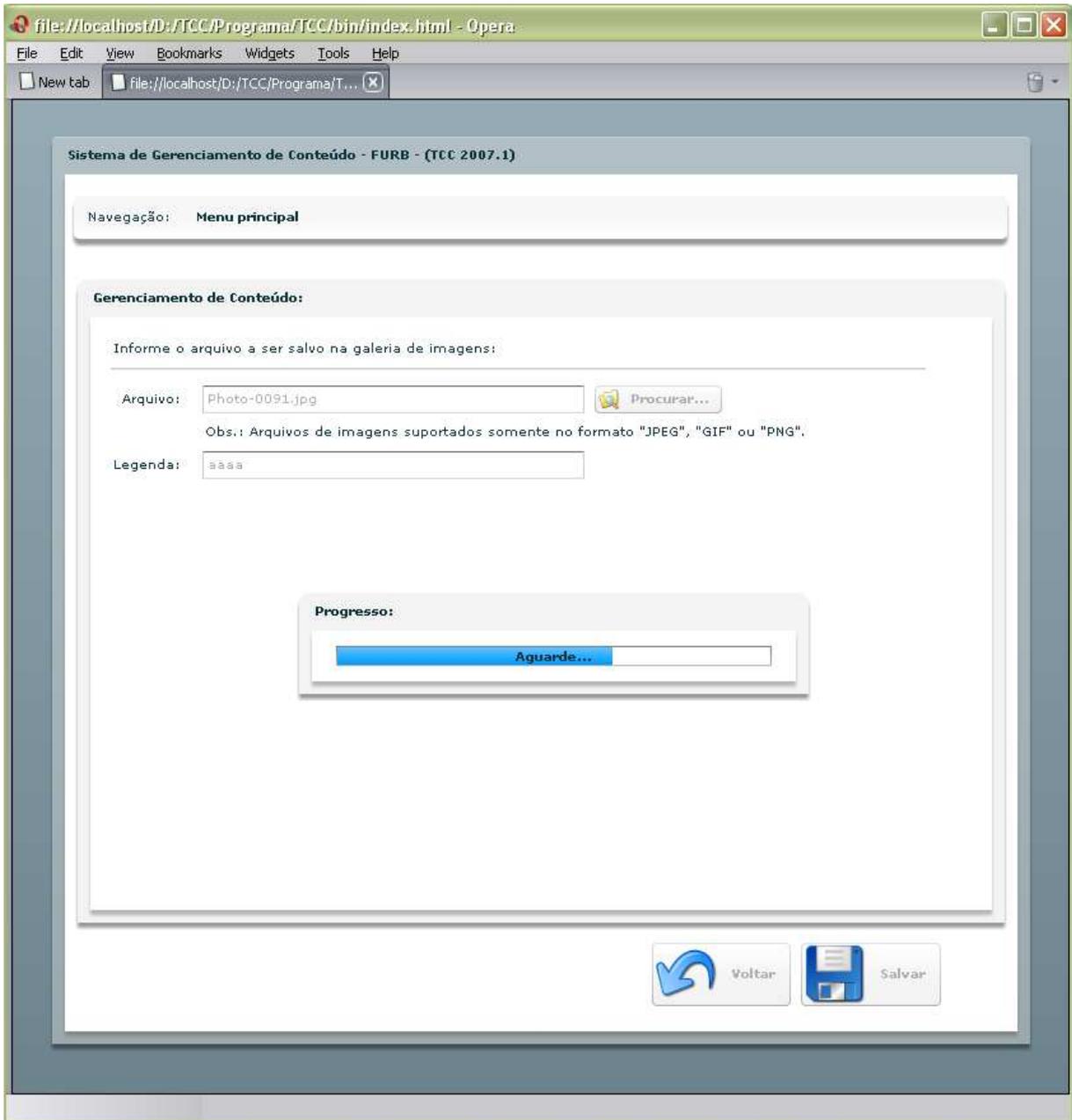


Figura 30 – Barra de progresso no envio do arquivo

É comum em aplicativo *web* apresentar campos para envio de arquivos, e quando o usuário clica no botão de enviar, ele fica sem um retorno visual do que está acontecendo. Tudo que ele sabe é que está sendo enviado o arquivo mas visualmente ele não sabe se deu algum problema, se parou de transmitir ou quanto do total do arquivo já foi transmitido. Essa é uma das características que o FLEX proporciona.

Ao término do envio do arquivo, o usuário é informado que foi enviado com sucesso e ele pode escolher outro arquivo, digitar uma nova legenda e enviar quantas imagens ele quiser. Quando terminar, ele irá possuir um conteúdo com texto formatado da forma que ele gostaria de apresentar e com uma galeria de imagem, se assim desejar.

Acessando o *website* pela página inicial do mesmo, o usuário terá as informações

cadastradas apresentadas no sistema de gerenciamento de conteúdo sem maiores problemas. Esta visualização do caso de uso pode ser vista na Figura 31.

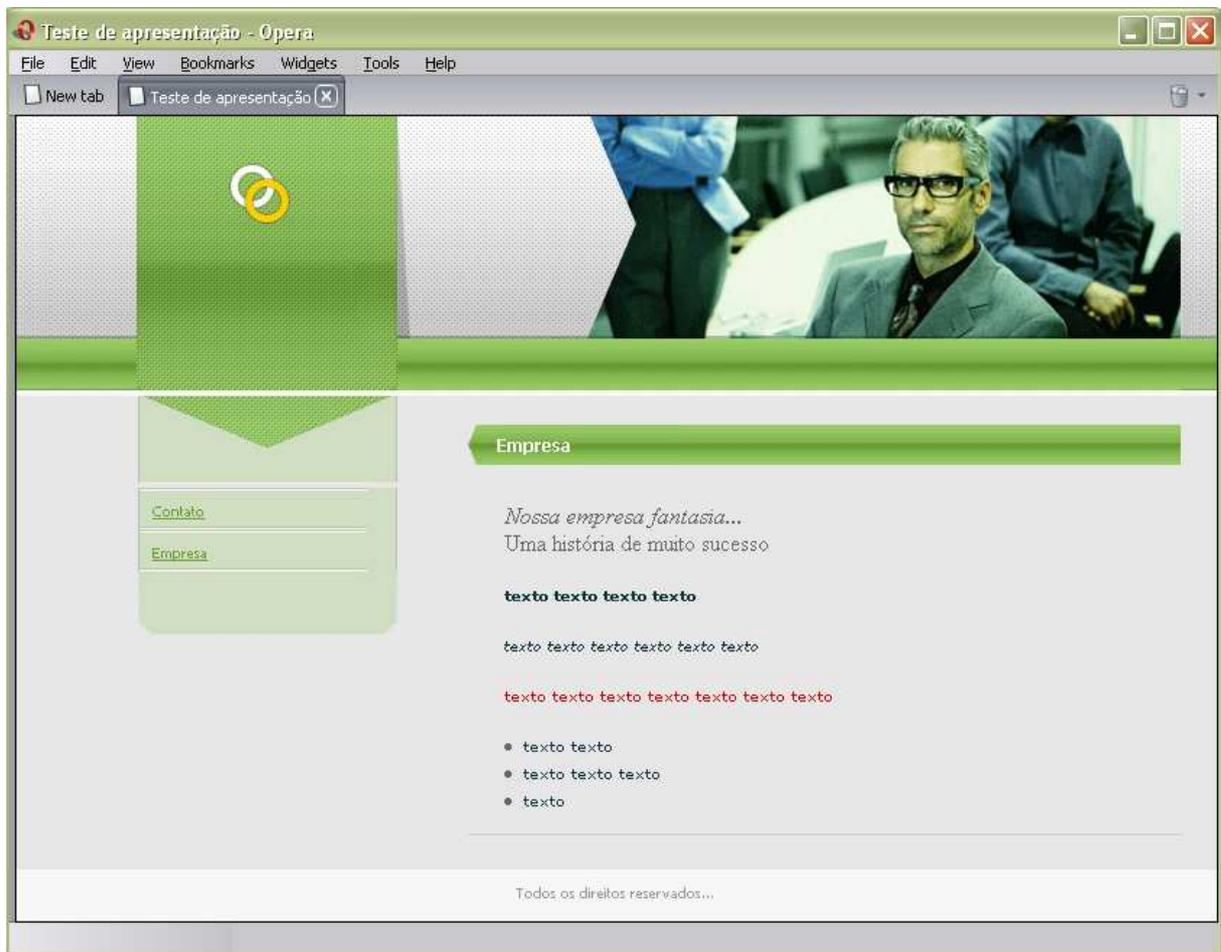


Figura 31 – Visualização do *website* com os conteúdos cadastrados

3.4 RESULTADOS E DISCUSSÃO

Um sistema de gerenciamento de conteúdo pode ser programado utilizando inúmeras linguagens de programação e apresentado de várias formas diferentes, seja como um aplicativo *desktop*, uma página web ou um aplicativo na web como foi apresentado aqui. Entretanto, a apresentação e a representação visual dos acontecimentos, seja cadastrando algum item de menu, um conteúdo ou até mesmo enviando um arquivo é muito importante para o usuário.

Com relação aos trabalhos correlatos e a tecnologia utilizada, o sistema desenvolvido manteve as características inerentes a aplicativos RIA. Um exemplo é o sistema ser carregado

apenas uma vez pelo usuário e logo após ele está livre para utilizá-lo sem interrupções. As únicas pausas existentes no sistema são os acessos ao banco de dados, seja para consulta ou para salvar informações, entretanto, a interface nunca é transmitida novamente para o usuário, apenas os dados trafegam durante o uso da aplicação.

Observando os outros sistemas de gerenciamento de conteúdo, a grande vantagem do sistema desenvolvido fica relacionado a sua interface com o usuário, pois além de nunca fazer parte da transmissão de dados durante o uso da aplicação, o que é muito comum em sistemas para a web, ela também possui a vantagem de responder em tempo real e fornecer ao usuário o respaldo necessário para saber o que está acontecendo em cada ação tomada por ele.

Comparado com aplicativos RIA desenvolvidos em AJAX, a organização do código fonte se torna muito mais simples e fácil de utilizar. As técnicas de desenvolvimento no AJAX são muito eficientes manipulando os dados e transmissões assíncronas, entretanto a utilização do JavaScript como linguagem torna a organização um problema, não só em termos de código fonte mas também na execução das tarefas.

Com o FLEX, a velocidade de execução do aplicativo é muito superior quando comparada a uma linguagem de *script* interpretada pelo navegador em vista que o código é compilado e executado em linguagem de máquina. É comum desenvolvedores que utilizam o AJAX como técnica de desenvolvimento sofrerem com problemas de escopo de variável e falta de consistência de tipos de variáveis durante a implementação, o que não ocorre com o FLEX. Muitos necessitam utilizar *frameworks* separados que cuidam somente das classes responsáveis pelo AJAX.

Com relação ao uso do ErgoList, o Quadro 9 apresenta o *checklist* informando se o item se aplica ao sistema desenvolvido ou não, justificando cada um dos itens.

| ErgoList |
|--|
| <p>Presteza</p> <p>Atende. O sistema conduz o usuário utilizando janelas padrões, com campos posicionados da mesma maneira na apresentação dos formulários contendo elementos específicos para a entrada de dados.</p> |
| <p>Agrupamento por localização</p> <p>Atende parcialmente. Os painéis e agrupamentos de campos e formulários seguem uma ordem lógica e similar em todas as telas apresentadas para o usuário. Os botões não possuem</p> |

teclas de atalho para executar determinada função mais rapidamente, como salvar os dados por exemplo.

Agrupamento por formato

Atende. Caixas de agrupamento são utilizadas para separar a interface e formulários, cursores animados são utilizados para indicar a situação atual do sistema, como o cursor de “ocupado” quando o sistema estiver carregando ou salvando alguma informação e apresenta mensagens de alertas em destaque para o usuário.

Feedback

Atende parcialmente. O usuário é informado quando está carregando ou salvando os dados e também fornece uma barra de progresso no envio das imagens para o sistema, mas o não fornece informações de tempo estimado para conclusão dos processos, bem como histórico dos eventos e processos desempenhados.

Legibilidade

Atende. Os campos de formulários possuem altura mínima e destaque para campos com conteúdos maiores, os textos estão legíveis e evitam utilizar negrito e sublinhado para não sobrecarregar visualmente o aplicativo, além de utilizar ícones maiores e mais destacados com legendas textuais.

Concisão

Atende parcialmente. Os ícones utilizados possuem tamanho maior do que o normal em aplicativos *desktop*.

Ações Mínimas

Atende parcialmente. O sistema não posiciona o cursor no primeiro campo dos formulários para preenchimento.

Densidade Informacional

Atende. As janelas são apresentadas nas mesmas posições, visando evitar a distribuição de objetos distintos na tela e apresenta somente as informações necessárias para a conclusão das funções requisitadas.

Ações Explícitas

Atende parcialmente. O sistema atende este item parcialmente pois os formulários não aceitam a tecla “ENTER” como uma tecla de ação de envio do formulário, e ela não possui função específica durante o preenchimento dos formulários.

Controle do Usuário

Atende parcialmente. O sistema permite cancelar e retornar os menus durante o preenchimento dos formulários, mas não permite o cancelamento do envio de uma imagem.

Flexibilidade

Não atendido. O sistema não permite customização da aparência e/ou janelas apresentadas.

Experiência do Usuário

Atende parcialmente. O usuário pode se deslocar de uma opção de menu para outra rapidamente, no entanto ele não oferece formas variadas de apresentar as mesmas informações aos diferentes tipos de usuário, seja avançado ou leigo.

Proteção contra erros

Atende parcialmente. O sistema não faz uso de sinais sonoros para eventuais erros ocorridos.

Mensagens de erro

Atende parcialmente. As mensagens de erros são simples e informam os eventuais problemas que possam ocorrer, entretanto não ajudam a solucionar em caso de algum erro grave e/ou informa como prosseguir e conseguir ajuda.

Correção de erros

Atende parcialmente. O usuário pode alterar os textos, apagar ou inserir, mas não há como restaurar textos de conteúdo ou menus cadastrados no sistema uma vez que os mesmos sejam excluídos.

Consistência

Atende. Os rótulos, textos e posicionamentos dos formulários e do sistema como um todo são apresentados na mesma posição mantendo a consistência, bem como as mensagens de erros nos campos são informadas com a cor vermelha destacando o campo.

Significados

Não atende. Os títulos das páginas no topo do aplicativo não reflete a última escolha feita pelo usuário.

Compatibilidade

Não aplicável.

Quadro 9 – Checklist ErgoList e seus respectivos itens

4 CONCLUSÕES

Este trabalho se propôs a desenvolver um sistema de gerenciamento de conteúdo utilizando uma nova linguagem de programação, conhecida por ActionScript 3, em um novo ambiente de desenvolvimento, FLEX, além de utilizar um novo conceito no desenvolvimento de software, que são os softwares na web, também conhecidos por RIA.

A utilização destes novos conceitos e tecnologias para desenvolver soluções na web se mostrou muito eficaz e promissor. A facilidade proporcionada pelo ambiente de desenvolvimento e a praticidade do sistema desenvolvido como aplicativo na web mostrou que existe muito espaço para este tipo de aplicação. Basta olhar alguns anos atrás, quando ler e-mails era algo para aplicações *desktop*.

Hoje há uma infinidade de serviços de e-mails que permitem a visualização e leitura dos mesmos no navegador do usuário, sem a necessidade da instalação de qualquer programa. Estes aplicativos mostram uma carência em sua usabilidade por herdarem características antigas e inerentes a linguagem HTML e JavaScript.

Desta forma o sistema desenvolvido apresenta uma alternativa para solucionar e apresentar aplicativos com interfaces mais intuitivas, mais poderosas e similares aos aplicativos *desktop*, possuindo velocidade de execução similar e com respostas instantâneas aos comandos do usuário.

Os objetivos de desenvolver um sistema de gerenciamento de conteúdo para web utilizando conceito de RIA com o FLEX foram alcançados, trazendo maiores facilidades no uso do aplicativo, na interface, maior rapidez na execução das tarefas e redução do uso dos servidores de hospedagem e tráfego de dados, transmitindo apenas os dados requisitados e não dados relacionados à interface do sistema como ocorre em aplicativos tradicionais na web, agilizando desta forma o processo como um todo.

O sistema proposto poderia ter sido desenvolvido utilizando várias outras linguagens de programação existentes, entretanto, o uso de novas e promissoras tecnologias apresentam inovações e apontam possíveis direções que os aplicativos web podem tomar. Quem sabe até a convergência de muitos aplicativos de escritório para aplicativos na web, como editores de texto e planilhas de dados.

No entanto, o trabalho desenvolvido possui algumas limitações, como a impossibilidade de se utilizar múltiplos usuários no sistema, designar tarefas para serem aprovadas e/ou corrigidas caso necessário, inserir vídeos nos conteúdos ou arquivos para

downloads e sistema para gerenciamento de *banners*. Outra limitação da aplicação é a impossibilidade de visualizar as estatísticas do *website*, informando as páginas de conteúdos mais acessadas, resolução de vídeo dos usuários que acessam o conteúdo, navegadores mais utilizados e até mesmo *links* de referencia que apontam para o *website*.

Sistemas de gerenciamento de conteúdo estão e vivem em constantes atualizações. As novas tecnologias aparecem a todo instante, e novas manias surgem a todo o momento. Muitos recursos disponíveis nos sistemas de gerenciamento de conteúdo observados nos trabalhos correlatos, tais como, gerenciamento de banners, relatórios de acessos, controle de usuários, sistema de busca interno, não estão disponíveis no sistema desenvolvido. O foco do trabalho era apresentar uma solução diferente e inovadora para um problema existente, este objetivo foi alcançado. Possíveis adições que complementariam o sistema são detalhadas na seção extensões.

Este trabalho foi importante para que se pudesse demonstrar e aplicar novas tecnologias, como o FLEX, e novos conceitos, como o RIA. Acredita-se que durante o desenvolvimento do trabalho, conhecer e aplicar estas tecnologias e conceitos na prática seja de grande importância para o desenvolvimento de futuras aplicações na web, possibilitando maiores inovações e funcionalidades nos aplicativos desenvolvidos.

4.1 EXTENSÕES

Este trabalho apresenta apenas as funções básicas de um sistema de gerenciamento de conteúdo. Assim sendo, existem muitas outras adições que poderiam ser feitas neste trabalho.

Como sugestões de extensão deste trabalho poderiam ser adicionadas e/ou aprimoradas as seguintes funcionalidades:

- a) melhoria na segurança utilizando encriptação dos dados enviados e recebidos;
- b) implementar a possibilidade de alterar a ordem dos menus no sistema;
- c) permitir a alteração da página inicial do *website* com conteúdo próprio;
- d) adicionar um sistema de *banners* rotativos ou de acordo com o assunto da página;
- e) adicionar um controle de usuários com níveis de acesso e permissões para inserir, aprovar e liberar novas adições na página, seja de menu ou conteúdo;
- f) elaborar estatísticas de acesso individual em cada página;
- g) criar uma área destinada a modificação e/ou criação de *templates*.

REFERÊNCIAS BIBLIOGRÁFICAS

ADOBE SYSTEMS INCORPORATED. **FLEX 2 technical white paper**. [S.l.], 2006a.

Disponível em:

<http://www.adobe.com/products/flex/whitepapers/pdfs/flex2wp_technicaloverview.pdf>.

Acesso em: 23 ago. 2006.

_____. **Adobe Flash Player version penetration**. [S.l.], [2006b]. Disponível em:

<http://www.adobe.com/products/player_census/flashplayer/version_penetration.html>.

Acesso em: 23 ago. 2006.

_____. **Macromedia Flash Player getting started with Flash**. [S.l.], [2006c]. Disponível em:

<http://download.macromedia.com/pub/documentation/en/flash/fl8/fl8_getting_started.pdf>.

Acesso em: 15 out. 2006.

_____. **Adobe**. [S.l.], 2007. Disponível em: <<http://www.adobe.com/>>. Acesso em: 15 maio 2007.

BROWNING, Paul; LOWNDES, Mike. **JISC TechWatch Report: content management systems**. [S.l.], 2001. Disponível em: <http://www.jisc.ac.uk/uploaded_documents/tsw_01-02.pdf>. Acesso em: 25 out. 2006.

CAMPBELL, Glen. **Siteframe**. [S.l.], 2005. Disponível em: <<http://siteframe.org/>>. Acesso em: 03 set. 2006.

CARDOZO, André. Site dinâmico na hora. **Info Exame**, São Paulo, n. 243, p. 78-80, jun. 2006.

CONVERSE, Tim; PARK, Joyce. **PHP a bíblia**. Tradução Edson Furmankiewicz. Rio de Janeiro: Campus, 2001.

CRANE, David; PASCARELLO, Eric. **AJAX in action**. Greenwich: Manning, 2006.

CYBIS, Walter A. (coord.). **Projeto ErgoList**. Florianópolis, 1998. Disponível em:

<<http://www.labiutil.inf.ufsc.br/ergolist/>>. Acesso em: 26 out. 2006.

DOSSIÉ SOBRE O MERCADO DE DOMÍNIOS NA INTERNET. [S.l.]: VeriSign, v. 2, n. 4, nov. 2005. Disponível em: <<http://www.verisign.com.br/static/032927.pdf>>. Acesso em: 17 ago. 2006.

ECLIPSE. **Eclipse.org home**. [S.l.], 2007. Disponível em: <<http://www.eclipse.org/>>.

Acesso em: 11 maio 2007.

EICHORN, Joshua. **Understanding AJAX: using JavaScript to create rich internet applications**. [S.l.]: Prentice Hall, 2006.

FABULOUS FORCE DATABASE TOOLS. **DBDesigner**. [S.l.], 2003. Disponível em: <<http://www.fabforce.net>>. Acesso em: 09 mar. 2007.

FAIN, Yakov; RASPUTNIS, Victor; TARTAKOVSKY, Anatole. **Rich Internet Application with Adobe Flex and Java**. [S.l.], 2006. Disponível em: <http://java.sys-con.com/read/210991_1.htm>. Acesso em: 17 out. 2006.

GEORGI, Oliver. **WCMS**. [S.l.], 2002. Disponível em: <<http://www.phpwcms.de/>>. Acesso em: 03 set. 2006.

GOOGLE. **Google Maps**. [S.l.], 2006. Disponível em: <<http://maps.google.com/>>. Acesso em: 16 out. 2006.

JOOMLA. **Joomla!** [S.l.], 2006. Disponível em: <<http://www.joomla.org/>>. Acesso em: 03 set. 2006.

MAMBO. **Mambo Server**. [S.l.], 2000. Disponível em: <<http://www.mamboserver.com/>>. Acesso em: 03 set. 2006.

MCLELLAN, Drew. **Very dynamic web interfaces**. [S.l.], 2005. Disponível em: <<http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>>. Acesso em: 15 out. 2006.

MILLARCH, Francisco. **O que é um CMS e porque você precisa de um**. [S.l.], 2005. Disponível em: <<http://webinsider.uol.com.br/index.php/2005/06/08/o-que-e-cms-e-porque-voce-precisa-de-um/>>. Acesso em: 14 out. 2006.

NIELSEN, Jakob. **Usability 101: fundamentals and definition – what, why, how**. [S.l.], 2006. Disponível em: <<http://www.useit.com/alertbox/20030825.html>>. Acesso em: 26 out. 2006.

PATRICK, Jeff. **A nonprofit's guide to content management systems**. [S.l.], 2006. Disponível em: <http://www.commonknow.com/reports/Common_Knowledge_Selecting_a_CMS.pdf>. Acesso em: 31 mar. 2007.

PHPMYADMIN. **phpMyAdmin**. [S.l.], 2007. Disponível em: <<http://www.phpmyadmin.net/>>. Acesso em: 11 maio 2007.

PNSF. **Postnuke**. [S.l.], 2001. Disponível em: <<http://www.postnuke.com/>>. Acesso em: 03 set. 2006.

RICH INTERNET APPLICATION. In: WIKIPEDIA, the free encyclopedia. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <http://en.wikipedia.org/wiki/Rich_internet_application>. Acesso em: 23 ago. 2006.

SPARX SYSTEMS. **Enterprise Architect**. [S.l.], 2007. Disponível em: <<http://www.sparxsystems.com>>. Acesso em: 07 mar. 2007.

SYNODINOS, Dionysios G. **CMS in a nutshell**. [S.l.], 2005. Disponível em: <<http://www.unixreview.com/documents/s=9900/ur0510h/ur0510h.htm>>. Acesso em: 14 out. 2006.

TAPPER, Jeff et al. **Adobe Flex 2: training from the source**. [S.l.]: Adobe Press, 2006.

WALSH, Norman. **A technical introduction to XML**. [S.l.], 1998. Disponível em: <<http://www.xml.com/pub/a/98/10/guide0.html>>. Acesso em: 30 mar. 2007.

YAHOO. **Yahoo! Maps beta**. [S.l.], 2006. Disponível em: <<http://maps.yahoo.com/beta/>>. Acesso em: 3 set. 2006.