

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

DESENVOLVIMENTO DE UM SISTEMA DE
GERENCIAMENTO DE CONTEÚDO PARA WEB
UTILIZANDO PHP5 E MYSQL

JEAN CARLOS BORMANIERI

BLUMENAU
2005

2005/1-24

JEAN CARLOS BORMANIERI

**DESENVOLVIMENTO DE UM SISTEMA DE
GERENCIAMENTO DE CONTEÚDO PARA WEB
UTILIZANDO PHP5 E MYSQL**

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. Ricardo Alencar de Azambuja – Orientador

**BLUMENAU
2005**

2005/1-24

**DESENVOLVIMENTO DE UM SISTEMA DE
GERENCIAMENTO DE CONTEÚDO PARA WEB
UTILIZANDO PHP5 E MYSQL**

Por

JEAN CARLOS BORMANIERI

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

Presidente: _____
Prof. Ricardo Alencar de Azambuja – Orientador, FURB

Membro: _____
Prof. Fábio Rafael Segundo – FURB

Membro: _____
Prof. Alexander Roberto Valdameri – FURB

Blumenau, 06 de julho de 2005

Dedico este trabalho a todas as pessoas que participaram diretamente ou indiretamente na realização deste, especialmente meus amigos e familiares.

AGRADECIMENTOS

À minha família, que sempre me incentivou nos estudos.

Aos meus amigos, por se fazerem sempre presentes nos momentos necessários.

Aos professores, por se esforçarem nos ensinamentos.

Ao meu orientador, Ricardo Alencar de Azambuja, por ter acreditado na conclusão deste trabalho.

As grandes idéias são aquelas nas quais a única coisa que nos surpreende é que não nos tivessem ocorrido antes.

Noel Clarasó, escritor espanhol.

RESUMO

Este trabalho de conclusão de curso especifica e implementa um sistema de gerenciamento de conteúdo para *web* utilizando PHP5 e MySQL, visando fornecer um mecanismo de disseminação de informações para a *internet*.

Palavras-chave: Gerenciamento. Conteúdo. Internet.

ABSTRACT

This work presents specification and implementation of a content management system for web using PHP5 and MySQL, seeking to supply a mechanism of spread Internet information.

Key-Words: Management. Content. Internet.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação do funcionamento de um CMS	15
Figura 2 – Representação padrão de um campo textarea.	21
QUADRO 1 – Código necessário para carregar o HTMLArea.	22
Figura 3 – Representação de um campo textarea com o htmlArea.	23
Figura 4 – Funcionamento do RoboHelp	24
Figura 5 – Diagrama de Caso de Uso	27
Figura 6 – Diagrama de Classes	27
Figura 7 – Diagrama de atividades	28
Figura 8 – MER lógico	29
Figura 9 – MER físico	29
QUADRO 2 – Página principal do site	31
Figura 10 – Representação do funcionamento do sistema de <i>templates</i>	33
Figura 11 – Página de configuração do sistema	35
Figura 12 – Criação de conteúdo para o site	36
Figura 13 – Criação de seções para o site.....	36
Figura 14 – Gerenciamento de arquivos no site	37
Figura 15 – Representação do site com <i>template</i> padrão	38
Figura 16 – Representação do site com <i>template</i> simples	38

LISTA DE TABELAS

Tabela 1 – Características de Sistemas de gerenciamento de conteúdos	39
--	----

LISTA DE SIGLAS

API – Application Programming Interface

CMS – Content Management System

FTP – File Transfer Protocol

HTTP – Hypertext Markup Language

LDAP - Lightweight Directory Access Protocol

PHP – Personal Home Page

SSL – Secure Socket Layer

WYSIWYG – What You See Is What You Get

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS DO TRABALHO	13
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 SISTEMA DE GERENCIAMENTO DE CONTEÚDO	14
2.2 ANÁLISE DE FERRAMENTAS DE GERENCIAMENTO DE CONTEÚDO	16
2.2.1 Jetbox	16
2.2.2 Mambo	16
2.2.3 E107	17
2.3 PHP 5.....	18
2.3.1 Suporte robusto a programação orientada a objetos	18
2.3.2 Nova extensão de acesso a MySQL.....	19
2.3.3 Ferramentas de Interoperabilidade XML.....	20
2.4 HTMLAREA.....	21
2.5 ROBOHELP	24
3 DESENVOLVIMENTO DA FERRAMENTA	25
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	25
3.2 ESPECIFICAÇÃO	26
3.3 IMPLEMENTAÇÃO	30
3.3.1 Técnicas e ferramentas utilizadas.....	30
3.3.1.1 Sistema de gerenciamento	30
3.3.1.2 Templates.....	32
3.3.2 Operacionalidade da implementação	33
3.3.2.1 Estudo de caso – Criação de um site pessoal.....	33
3.4 RESULTADOS E DISCUSSÃO	39
4 CONCLUSÕES.....	40
4.1 EXTENSÕES	41
REFERÊNCIAS BIBLIOGRÁFICAS	42
APÊNDICE A – Página de administração – admin.php.....	43
APÊNDICE B – Página principal – index.php	45
APÊNDICE C – Página auxiliar de instalação – install.php	46

APÊNDICE D – Carregar arquivos para download – loadfile.php	48
APÊNDICE D – Listar arquivos de imagens – loading.php.....	49
APÊNDICE E – Efetuar logoff do sistema – logoff.php	51
APÊNDICE F – Classe Template – template.php	52
APÊNDICE G – Classe conteudo – conteudo.php	55
APÊNDICE H – Classe secao – secao.php	57
APÊNDICE I – Classe usuario – usuario.php	58

1 INTRODUÇÃO

Com o advento e o avanço da internet no cenário mundial, uma nova revolução vem acontecendo. Neste novo cenário, a informação está mais acessível a todos, o que permite atingir um público alvo maior. Seguindo esta linha de pensamento, é conveniente que pessoas e empresas publiquem e mantenham informações sobre seus produtos ou serviços, de modo que todos os possíveis interessados tenham alcance a estas informações na Internet.

Não há censura de conteúdo na internet. Sendo assim, qualquer pessoa interessada em qualquer tipo de assunto, pode publicar e manter informações de qualquer gênero. Segundo IBOPE (2004) estima-se que aproximadamente 20,5 milhões de pessoas acessem mensalmente a internet no Brasil, o que representa acima de 10% da população brasileira.

Segundo Abiteboul, Buneman e Suciú (2000, p. 2), “A *web* fornece um padrão simples e universal para a troca de informação”. Porém, os aplicativos existentes hoje que são destinados a manter e, atualizar informações na Internet de uma maneira geral, são desenvolvidos para usuários avançados ou administradores de redes, que já possuem certo conhecimento no assunto. Por outro lado, o usuário comum ou aquele que possui pouco conhecimento em informática, não consegue gerenciar tal procedimento.

Nesta condição, foi desenvolvida uma aplicação de gerenciamento de conteúdo para *web*, comumente conhecida como *Content Management Systems* (CMS), para auxiliar a elaboração e manutenção de *sites* na Internet destinado a usuários com pouco conhecimento nesta área.

O sistema foi desenvolvido utilizando a linguagem *Personal Home Page* (PHP) e banco de dados MySQL, devido ao grande número de *hosts* de hospedagem que oferecem estes recursos, visando assim facilitar a escolha de hospedagem pelo usuário

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é o desenvolvimento de um sistema de gerenciamento de conteúdo para web.

Os objetivos específicos do trabalho:

- a) desenvolver um sistema que possa ser executado em qualquer *host* de Internet que forneça suporte a linguagem PHP5 e banco de dados MySQL;
- b) fornecer mecanismos simples para que usuários criem, editem e mantenham informações em um *site* na Internet.

1.2 ESTRUTURA DO TRABALHO

A fundamentação teórica, descrita no capítulo 2 fornece uma visão geral sobre os sistemas de gerenciamento de conteúdo, falando um pouco sobre sua história e para qual problema esse sistema foi criado para resolver.

No capítulo 3, é apresentado o desenvolvimento da ferramenta, incluindo detalhes sobre a especificação e implementação realizada.

O capítulo 4 apresenta uma conclusão sobre o assunto, enfatizando os objetivos alcançados.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas informações sobre sistemas de gerenciamento de conteúdo, explorando a história, mapeando algumas de suas principais vantagens, apresentando algumas ferramentas de sistema de gerenciamento de conteúdo, analisando as principais vantagens e desvantagens dessas ferramentas.

2.1 SISTEMA DE GERENCIAMENTO DE CONTEÚDO

Os sistemas de gerenciamento de conteúdos, ou *Content Management Systems* (CMS), são ferramentas utilizadas para gerenciar a disseminação de informação através da internet. Elas são ferramentas que sistematizam e automatizam o processo de criação e atualização de informações em um *site*.

Este termo foi primeiramente utilizado em meados no ano de 1996. Apesar da grande quantidade de ferramentas atualmente no mercado, a idéia de gerenciamento de conteúdo era uma solução inovadora para os padrões da internet da época, visando resolver um dos grandes problemas emergentes da Internet. Como a *web* cresceu e sua complexidade aumentou, a necessidade de separação de conteúdo e *design* era inevitável.

Segundo HUNTER (2001), o crescimento e o aumento da complexidade tornaram necessária a separação de conteúdo e *design*, permitindo a possibilidade de apresentação da informação de acordo com a exigência do usuário. Atender a exigência dos usuários, é hoje, um dos grandes diferenciais no mercado.

Visando atender a necessidade de separar conteúdo do *design*, de uma maneira geral, um CMS é concebido de duas partes principais. O *template*, que segundo MORATELLI (2002) irá definir o formato gráfico como o site será apresentado, e um *script* que extrai o

conteúdo, geralmente armazenada em um banco de dados, apresentando-a conforme o *template* selecionado.

A figura 1 mostra uma representação do funcionamento de um CMS.

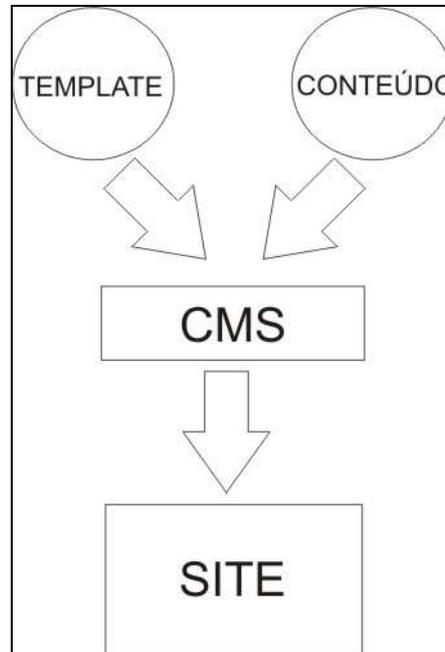


Figura 1 – Representação do funcionamento de um CMS

As principais vantagens de se utilizar um *CMS* para gerenciar um site são:

- c) transparência de localização: o usuário poderá criar ou editar os conteúdos, assim como as configurações do site de qualquer computador, através da internet;
- d) velocidade na atualização das informações: as informações são manipuladas diretamente no ambiente de desenvolvimento, agilizando o processo;
- e) facilidade de manutenção: o conteúdo está separado do *design* do site, o que permite a alteração de *layout* sem comprometer as informações já disponibilizadas;
- f) independência de conhecimento: qualquer usuário habilitado pode criar ou editar conteúdos, independentemente de conhecer alguma linguagem de programação para *web*. Essa vantagem barateia o custo de manutenção do site, pois não requer investimento em aprendizado de linguagens de programação.

2.2 ANÁLISE DE FERRAMENTAS DE GERENCIAMENTO DE CONTEÚDO

Atualmente existem diversas implementações de sistemas de gerenciamento de conteúdos. A seguir, serão citadas algumas das mais conhecidas, enfatizando suas principais características:

2.2.1 Jetbox

Segundo JETBOX (2004), o Jetbox é um sistema de gerenciamento de conteúdo muito utilizado na *web*. O que mais caracteriza essa ferramenta é sua interface de administração, que distribui organizadamente as funções da ferramenta.

Outra grande característica é seu *wizard*, que auxilia o usuário na instalação da ferramenta, fornecendo as informações necessárias, assim como mecanismos para efetuar tais configurações. Seu fluxo de trabalho é orientado a módulos, o que facilita o desenvolvimento de extensões para a ferramenta.

Apesar desta facilidade, o software tende a se tornar complexo para usuários que possuem pouco conhecimento em informática. A principal desvantagem desta ferramenta é a de privar os usuários com poucos conhecimentos de sua utilização.

2.2.2 Mambo

Diferentemente do Jetbox, o Mambo foi originalmente desenvolvido visando a facilidade de utilização. O desenvolvimento da ferramenta, segundo o site Mambo (2005), possui aproximadamente 300 pessoas organizadas em diversas equipes. Além disso, essa ferramenta possui um suporte técnico comercial, o que torna sua manutenção muito mais

freqüente do que a de outros sistemas de gerenciamento.

Uma grande característica desta ferramenta de gerenciamento de conteúdo é a de implementar e disponibilizar uma enorme quantidade de API's de desenvolvimento. Através delas, outros desenvolvedores poderão adaptar a ferramenta para atender seus problemas, ou até mesmo para que desenvolvam novas extensões para o aplicativo.

A desvantagem desta ferramenta é a de não fornecer suporte à ambientes Microsoft (*Windows + Internet Information Services*).

2.2.3 E107

Segundo E107.Org (2004), o sistema de gerenciamento de conteúdo E107 é equipado com uma interface distinta, baseada em blocos de informações. Originalmente voltado para portais de internet, esse sistema oferece diversas características que facilitam a administração de um canal onde haja grande volume de informações e de usuários.

Entre essas características, podemos citar o fórum integrado ao sistema que permite criar um canal contínuo de troca de informações entre os membros cadastrados. Existe também um poderoso gerenciador de artigos e *newsletters* que oferece um constante meio de disseminação de informações para quem faz parte dessas comunidades.

Apesar de ser uma ótima alternativa para sites de comunidades, ou portais de internet, a maior desvantagem da utilização deste sistema é sua complexidade. Faz-se necessário um grande conhecimento sobre o aplicativo para que se o utilize de maneira consistente.

2.3 PHP 5

PHP é uma linguagem de programação para criar *sites* dinâmicos na internet. Criada em meados de 1994, por Rasmus Lerdorf, seu desenvolvimento foi focado inicialmente em eficiência da linguagem, fornecendo uma grande gama de funcionalidades. Com a liberação da versão 4 (PHP 4) em 1998, houve um imenso ganho de performance com a inclusão do *engine* Zend. A partir daí que o PHP se popularizou e começou a ser amplamente utilizado para construir sites e ferramentas para a internet. A versão 4 foi sendo atualizada constantemente e ainda hoje é a mais utilizada.

Finalmente, após seis anos passados desde uma liberação de versão (*major release*) foi revisado e liberado o PHP 5, uma evolução necessária do PHP 4. Segundo TRACHTENBERG (2004), comparando com o PHP 4, a versão 5 possui melhoramentos em três principais áreas:

- a) programação orientada a objetos;
- b) MySQL;
- c) XML.

Estes itens foram completamente reescritos, transformando suas limitações em características principais. A seguir, esses itens são apresentados de uma maneira mais detalhada.

2.3.1 Suporte robusto a programação orientada a objetos

Desde o lançamento da primeira versão do PHP, programadores de todo o mundo solicitavam uma maior incorporação de característica da orientação a objetos. Entretanto, nem

na versão 3 assim como a versão 4 incluíram essas características nativamente.

A partir da versão 5, os programadores podem desfrutar de diversas propriedades a orientação a objetos. Entre elas, pode-se citar:

- a) construtores;
- b) destrutores;
- c) métodos e propriedades públicos, privados e protegidos;
- d) interfaces;
- e) classes abstratas;
- f) métodos e propriedades estáticos;
- g) métodos e classes finais.

Adicionalmente, os objetos são agora criados e passados por referência, em vez de por valor, facilitando o desenvolvimento de aplicações.

2.3.2 Nova extensão de acesso a MySQL.

Segundo CONVERSE e PARK (2001, p. 233), “...PHP e MySQL são ferramentas que caminham mutuamente juntas.”. Por serem aplicações *open-source* e mantidas por comunidades de desenvolvedores na internet, o uso das 2 ferramentas no desenvolvimento de aplicações para a internet é imenso.

As últimas versões do MySQL, 4.1 e 5.0 introduziram muitas novas características, algumas das quais requisitaram mudanças significantes na maneira como as extensões utilizadas para acessar este aplicativo funcionava. Como resultado, o PHP 5 vem com uma nova e completa extensão para acesso a banco de dados MySQL. Algumas das novas características são exemplificadas a seguir:

- a) declarações preparadas: são *queries* SQL que são pré-escritas e pré-compiladas no servidor de banco de dados;
- b) conexões ssl: permite que as conexões sejam criptografadas através do protocolo SSL;
- c) funções de múltiplas *queries*: permite a execução de múltiplas *queries* SQL todas de uma vez.

2.3.3 Ferramentas de Interoperabilidade XML

O PHP 5 corrige os maiores problemas com as extensões de acesso à XML da versão anterior. Enquanto que o PHP 4 permitia aos usuários manipular XML, suas ferramentas XML eram apenas superficiais. Cada uma dessas ferramentas cobria uma parte da estrutura do XML, mas elas não foram desenhadas para funcionarem juntas, tornando o suporte as características avançadas do XML fracas.

Essa interoperabilidade com o XML foi amplamente melhorada no PHP 5. As principais características incrementadas nas extensões de acesso XML são:

- a) todas trabalham em conjunto;
- b) são padronizadas em uma simples biblioteca XML;
- c) completamente padronizada com os padrões da W3;
- d) processamento de dados eficiente.

2.4 HTMLAREA

A criação de um sistema de gerenciamento de conteúdo envolve diversas características. Talvez a principal, é a de fornecer um meio compatível para que os usuários possam incluir e editar as informações em seu sistema. Em sistemas voltados a usuários com pouco conhecimento em informática, faz-se necessário que essa interação seja simples, ou seja, o usuário não precisa ter grandes conhecimentos para poder operar essas ferramentas.

Visando atender essa necessidade, começaram a surgir na *web* editores baseados em javascript, que podem ser incorporados aos formulários HTML, alterando a interface do campo `<textarea>` aprimorando-o para que se pareça como um editor HTML padrão.

A figura 2 mostra como é a representação de um campo `<textarea>` padrão de um formulário HTML.



Figura 2 – Representação padrão de um campo textarea.

Segundo BAZON (2003), o htmlArea é um editor *What You See Is What You Get*

(WYSIWYG) baseado em *javascript*, gratuito e customizável, que se incorpora aos campos `<textarea>` permitindo ao programador utilizá-lo em suas páginas HTML. Dentre suas principais características, se destacam:

- a) barra de ferramentas customizável;
- b) fácil tradução para outras linguagens;
- c) infraestrutura baseada em *plugins*;
- d) compatível com a especificação HTML da W3C;
- e) possui teclas de atalhos semelhantes ao do Microsoft Word;
- f) edição em modo tela cheia.

Por essas e outras características, o HTMLArea é uma das ferramentas mais utilizadas em sistemas de gerenciamento de conteúdos atualmente. Seu uso consiste na adição de algumas linhas de código à página HTML que será utilizada, conforme o quadro 1 representa:

```

<!-- Inicio HTMLAREA-->
<script language="JavaScript1.2">    <!-- // load htmlarea
_editor_url = "htmlarea/";           // URL to htmlarea files
var win_ie_ver = parseFloat(navigator.appVersion.split("MSIE")[1]);
if (navigator.userAgent.indexOf('Mac')      >= 0)
    { win_ie_ver = 0; }
if (navigator.userAgent.indexOf('Windows CE') >= 0)
    { win_ie_ver = 0; }
if (navigator.userAgent.indexOf('Opera')    >= 0)
    { win_ie_ver = 0; }
if (win_ie_ver >= 5.5) {
    document.write('<scr'+ 'ipt src="'+ _editor_url+ 'editor.js"');
    document.write(' language="JavaScript1.2"></scr'+ 'ipt>');
} else {
document.write('<scr'+ 'ipt>function editor_generate() ');
document.write('{ return false; }</scr'+ 'ipt>')}
// -->
</script>
<!-- Fim HTMLAREA-->

```

QUADRO 1 – Código necessário para carregar o HTMLArea.

A grande vantagem do HTMLArea em relação a outros editores é a de possibilitar que os criadores dos sistemas de gerenciamento de conteúdo adaptem a ferramenta a seus

sistemas.

O desenvolvimento do HTMLArea começou com a empresa Interactive Tools que após algumas versões lançadas, descontinuou a ferramenta. Porém, antes disso, a empresa liberou o código fonte sob a licença do tipo BSD, o que permitia que outros desenvolvedores continuassem a utilizá-la e a incorporar novas funcionalidades ao sistema.

Assim, o htmlArea continua sendo desenvolvido. Atualmente, o responsável pela manutenção do projeto é Mihai Bazon. O aplicativo está na versão 3.0.

A seguir, na figura 3, é representado um campo `<textarea>` onde o htmlArea está configurado para sobreescrevê-lo.

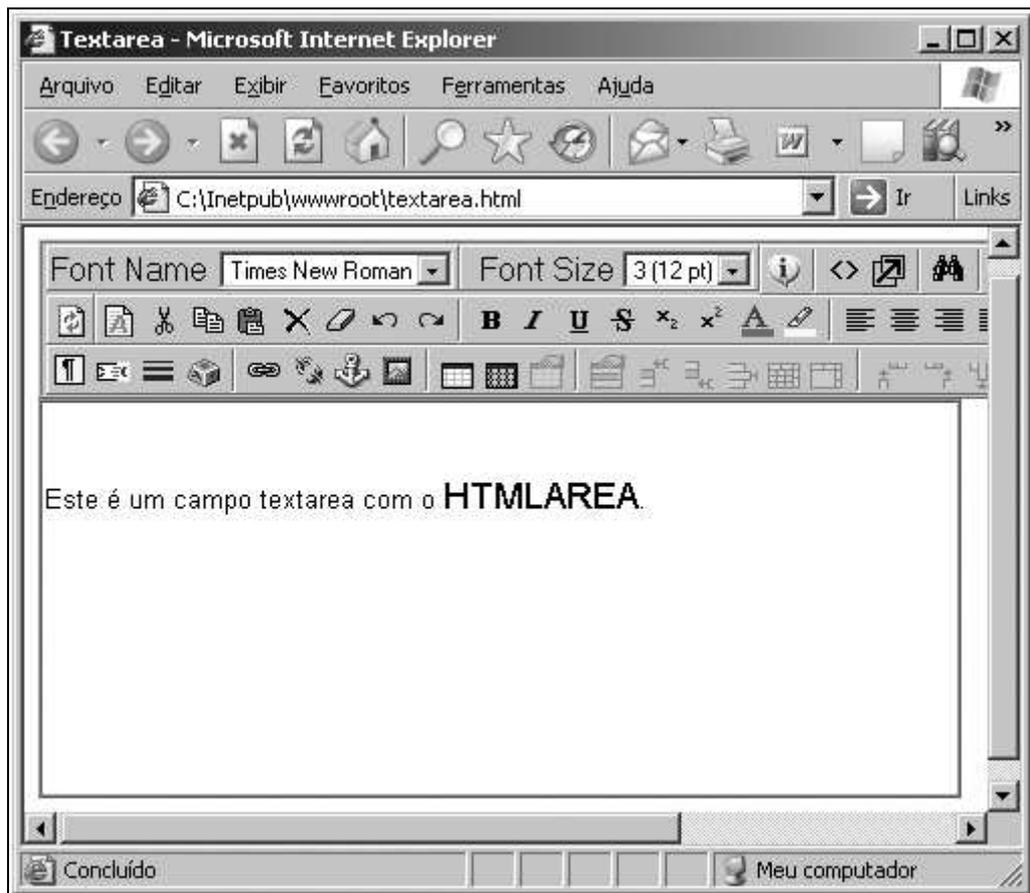


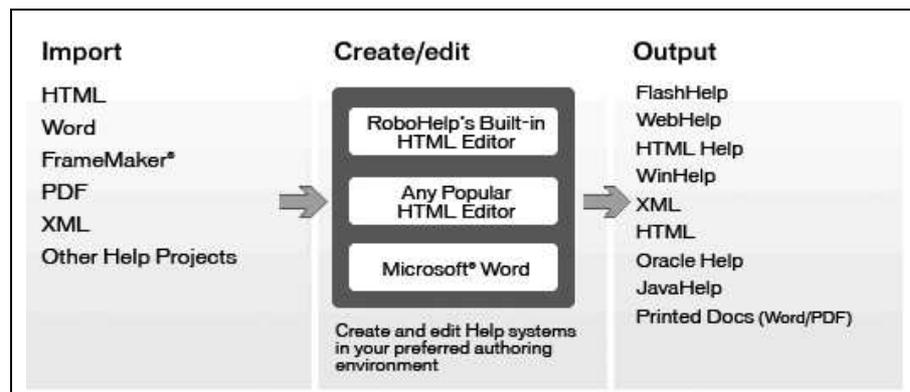
Figura 3 – Representação de um campo textarea com o htmlArea.

2.5 ROBOHELP

Segundo MACROMEDIA (2004), o RoboHelp Office é o mais fácil e rápido software de desenvolvimento de sistemas de ajuda e documentação, incluindo diversas características que auxiliam os usuários há facilmente criar seus documentos conforme suas necessidades.

O RoboHelp possui um excelente sistema de edição e criação de conteúdo, além da possibilidade de integração com os aplicativos da família *MSOffice*.

A figura 4 apresenta o funcionamento do RoboHelp.



Fonte: Macromedia (2004)

Figura 4 – Funcionamento do RoboHelp

Levando em conta a facilidade de uso e agilidade de desenvolvimento, o RoboHelp demonstra-se um excelente software para a elaboração de documentação de sistemas.

3 DESENVOLVIMENTO DA FERRAMENTA

O presente trabalho consiste na criação de um sistema que fornece mecanismos para que permitam aos usuários criar, editar e manter um site na internet. Atualmente, as ferramentas analisadas, são incompletas no que tange às necessidades dos usuários com pouco conhecimento em programação para internet. Além disso, os sistemas de gerenciamento de conteúdo analisados não oferecem compatibilidade ao PHP 5. Esta versão do PHP possui muitos novos avanços e tende a se tornar padrão dentro de algum tempo.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O trabalho consiste em um sistema de gerenciamento de conteúdo. Como este sistema será executado *on-line*, deverá ser implementado um processo para a autenticação dos usuários, criando uma camada de segurança, permitindo assim que somente usuários válidos possam fazer alterações no sistema.

Os usuários autenticados podem alterar as configurações padrão do site, conseguindo assim modificar o modo como ele será representado ao ser acessado. Essas alterações deverão ser efetuadas em uma tela diferente da tela principal do site.

Depois de autenticados, os usuários podem criar novos conteúdos ou seções, além de editá-los e publicá-los. Existe um editor de conteúdo *on-line*, semelhante a um editor html que auxiliará o usuário nesta tarefa.

O usuário pode efetuar *upload* de arquivos e imagens. Isso eliminará a necessidade de atualização de arquivos no site por outro meio, centralizando assim a manutenção do conteúdo através da interface do sistema.

O usuário pode criar *links* diretamente para os documentos que são copiados através de

upload, facilitando a distribuição de arquivos.

O sistema implementa suporte a *templates*, podendo assim o administrador alterar a maneira como o site é representado apenas alternando a configuração deste template.

O sistema pode ser executado nos principais *hosts* de internet, desde que tenham suporte a PHP 5 e MySQL. O usuário necessita apenas saber detalhes de configurações de acesso ao banco de dados e de sua conta de acesso ao servidor, sendo esses dados geralmente passados ao usuário quando este assina um plano de hospedagem com as empresas de hospedagem de *sites*.

Ao ser executado pela primeira vez, o sistema orientará o usuário a fazer os primeiros passos da configuração do sistema. Nesta etapa, serão configuradas informações de acesso ao banco de dados e a página de administração do sistema.

3.2 ESPECIFICAÇÃO

Esta seção apresenta as especificações utilizadas para o desenvolvimento do sistema de gerenciamento de conteúdo. Para a criação dos modelos e diagramas, foi utilizada a ferramenta *Sybase Power Designer*.

O sistema foi desenvolvido utilizando a orientação a objetos. Essa metodologia foi empregada para verificar a integridade dessa metodologia no PHP 5, que segundo seus desenvolvedores, melhorou o suporte a orientação a objetos.

A figura 5 apresenta o diagrama de caso de uso do sistema. O administrador, após se autenticar, faz as configurações necessárias para a utilização do *site* e faz a publicação dos conteúdos necessários.

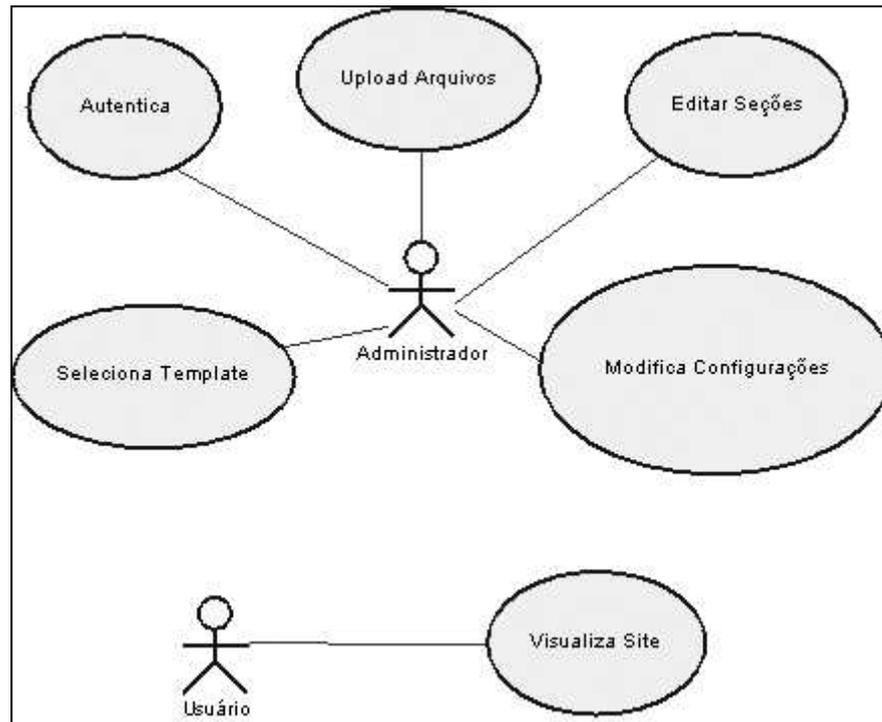


Figura 5 – Diagrama de Caso de Uso

O usuário, por sua vez, faz a solicitação de acesso ao site e recebe a página processado conforme o administrador definiu através das configurações e do conteúdo publicado.

A figura 6 representa o diagrama de classes do sistema de gerenciamento de conteúdo e a maneira como as classes interagem entre si.

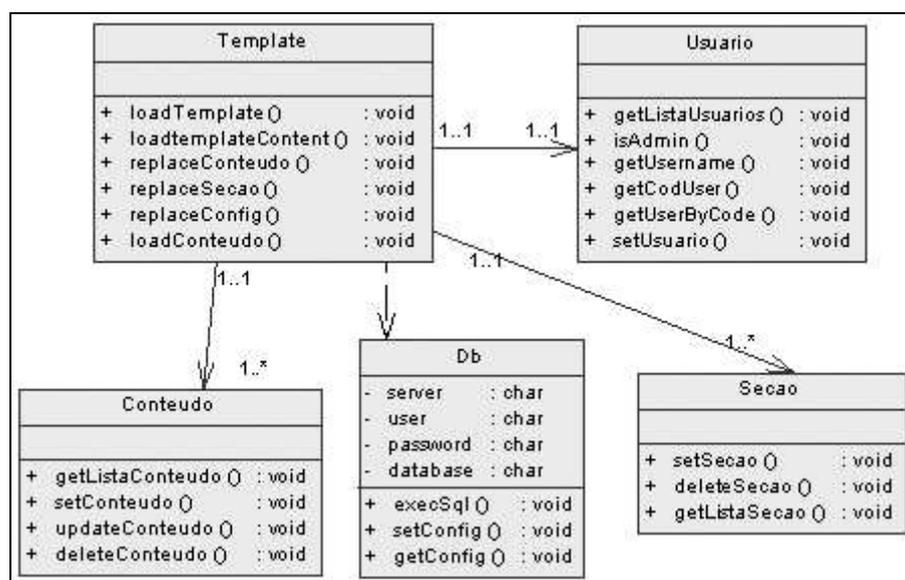


Figura 6 – Diagrama de Classes

A principal classe do sistema é a “Template”. Essa classe é instanciada na página principal e após ser instanciada, é feita a chamada ao método “LoadTemplate”, que será responsável por carregar o *template* configurado no banco de dados e então montar a página principal do *site*.

O resultado final do processo é a criação da página *web*, baseado nas configurações, conteúdos e seções configurados pelo administrador do *site*. O processo é apresentado no diagrama de atividades da figura 7.

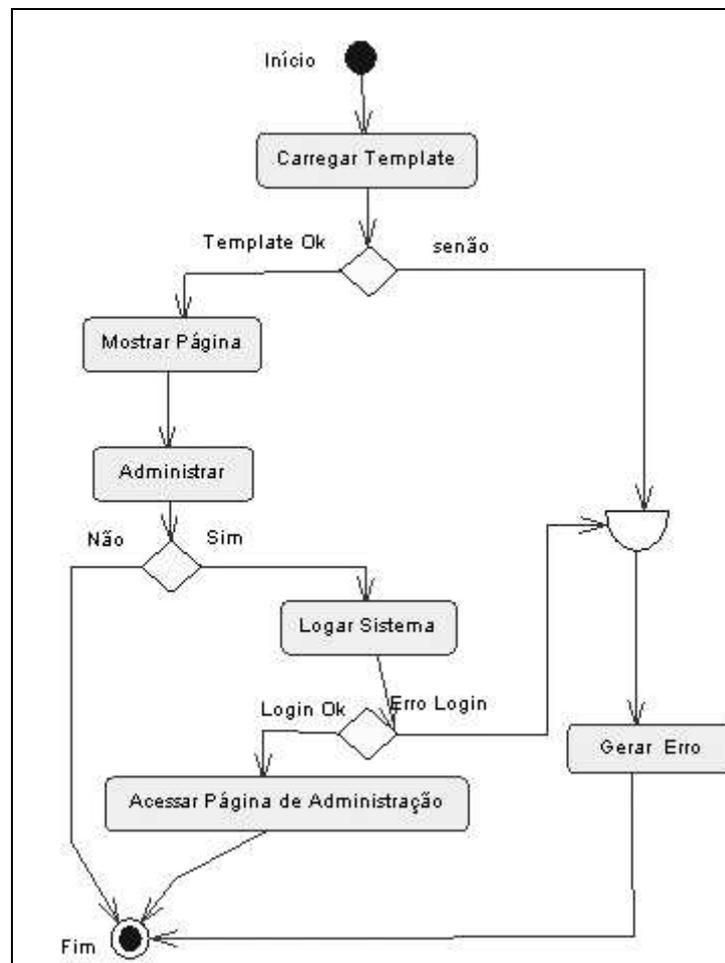


Figura 7 – Diagrama de atividades

Ao acessar a página, é carregado o *template*, onde caso haja algum problema, é gerado um erro. Se não houver erros, é apresentada a página do site. O usuário pode acessar a página de administração, onde será solicitado seu login e senha. Se o login ou senha estiverem incorretos, será apresentado uma mensagem de erro, senão será apresentada a página de

administração onde o usuário poderá efetuar as alterações necessárias.

Os dados do sistema são armazenados em um banco de dados MySQL. A figura 8 apresenta o MER lógico que indica todas as variáveis para o armazenamento das informações do site.

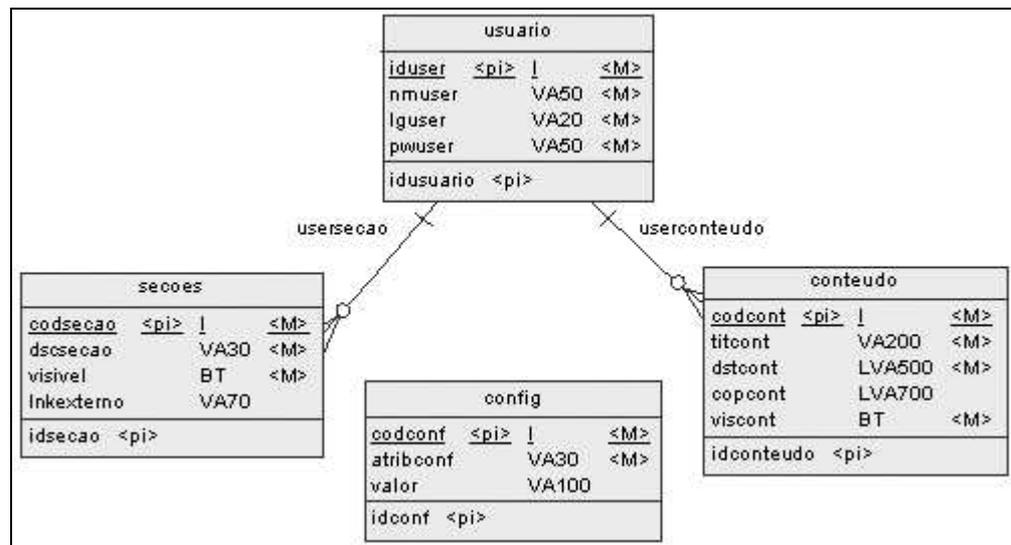


Figura 8 – MER lógico

A partir do MER lógico, é gerado o MER físico que está representado na figura 9, o qual indica as ligações entre as entidades.

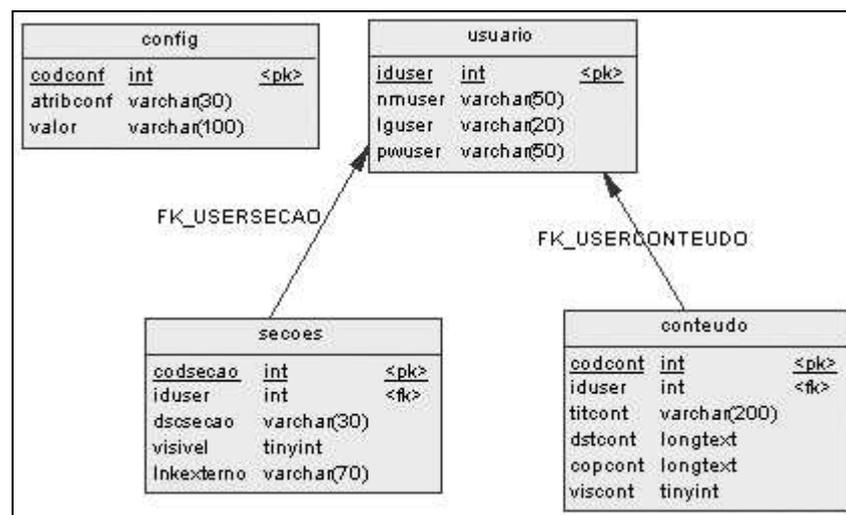


Figura 9 – MER físico

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas informações sobre a implementação do trabalho expondo as técnicas e ferramentas utilizadas na implementação.

3.3.1 Técnicas e ferramentas utilizadas

O sistema de gerenciamento de conteúdo para *web* desenvolvido é uma ferramenta que permite aos usuários criar e manter um site na *web* de maneira simples. Essa facilidade é obtida por fornecer um mecanismo onde o usuário não precise possuir nenhum conhecimento de programação. Para implementação do sistema, foi utilizado o PHP 5. Essa escolha foi feita, porque os sistema de gerenciamento pesquisados não são compatíveis com esta plataforma de desenvolvimento.

A seguir são apresentadas as principais técnicas do sistema. Primeiramente, será abordado o sistema de uma maneira geral e após, será efetuado uma análise sobre como funciona a integração dos *templates* ao sistema.

3.3.1.1 Sistema de gerenciamento

O sistema consiste de uma série de *scripts* feitos em PHP que executam em um servidor *web*. Quando uma requisição é feita ao servidor, esse procura pelo arquivo *index.php* iniciando o processo de construção da página. Esse início, dá-se no carregamento da função *LoadTemplate* de um objeto *Template* instanciado. Esta função então carrega em memória os arquivos do *template* configurado através do banco de dados para utilizá-lo na montagem do site.

Com o *template* em memória, o sistema começa a montar a página carregando as seções e conteúdos disponibilizados previamente pelo administrador do site para serem exibidos. As informações, depois de recuperadas, são dispostas conforme determinadas pelo arquivo de *template*. No quadro 2 é apresentada a implementação da página principal.

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : index.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 31/05/2005
*
*****
* Função    : Mostrar a página Inicial
*****/

// Verificar se as configurações iniciais já foram efetuadas,
// senão, redirecionar para a página de instalação.
$dir_install = 'install.php';

if (file_exists($dir_install)) {
    header('location:'.$dir_install);
} else {
    //Inclui arquivo de definição de classes
    include('classes/db.php'); include('classes/template.php');

    //Criar uma conexão com o banco de dados para buscar informações do banco
    $banco = new db;

    //Ler as configurações do banco de dados para saber qual template será carregado
    $template_dir = 'templates/'.$banco->getConfig("templatesite");

    $template = new template;

    if (isset($_HTTP_GET_VARS["content"])){
        $template->loadTemplateContent($template_dir, $_HTTP_GET_VARS["content"]);
    }else{
        $template->loadTemplate($template_dir);
    }
}
?>

```

QUADRO 2 – Página principal do site

3.3.1.2 Templates

Os *templates* determinam como as informações extraídas do banco de dados serão apresentadas no site. A escolha do *template* é feita através da página de administração.

Na estrutura de diretórios do sistema, a pasta /templates contém os *templates* lidos pela aplicação. Cada sub-diretório desta pasta, é tratado como sendo um *template* utilizável pelo sistema.

O usuário pode criar seu próprio *template* para adaptar a interface do *site* à sua maneira, ou escolher um dos padrões já existentes. A estrutura básica para a construção de um *template*, necessita de três arquivos padrões

- a) *conteudo.tpl*: este arquivo determina como o conteúdo será apresentado no site. Os conteúdos são informações que estarão presentes na página principal do site;
- b) *main.tpl*: este arquivo determina a maneira como o site será apresentado de uma maneira geral, sendo um arquivo html com algumas informações adicionais;
- c) *secao.tpl*: este arquivo determina como as seções serão apresentadas no site. As seções são os itens que aparecem no menu.

Os *templates* possuem palavras reservadas que são substituídas quando ele é carregado. Dessa maneira, o script que monta a página que será apresenta carrega o *template* e incorpora as informações que serão apresentadas, substituindo as palavras chaves do *template*. Na figura 10, é apresentada o funcionamento deste processo.

1 <html>	1 <html>
2 <head>	2 <head>
3 <title>	3 <title>
4	4
5 {TITULOSITE}	5 Teste de Título
6	6
7 </title>	7 </title>
8 </head>	8 </head>
9 <body topmargin="0" leftmargin="0">	9 <body topmargin="0" leftmargin="0">
10 Somente título.	10 Somente título.
11 </body>	11 </body>
12 </html>	12 </html>

Figura 10 – Representação do funcionamento do sistema de *templates*

O quadro a esquerda representa o arquivo `main.tpl`, o qual é responsável pela estrutura da página. O quadro a direita, mostra a página processada, utilizando-se este arquivo de *template*. Podemos observar na linha 5 que a palavra reservada `{TITULOSITE}` foi substituída pelo valor correspondente ao título do site que estava gravado no banco de dados no momento da execução do sistema.

3.3.2 Operacionalidade da implementação

Utilizando o sistema desenvolvido, é possível criar e manter sites na internet. Nesta seção é apresentado um estudo de caso visando mostrar as funcionalidades do sistema, para a construção de uma página.

3.3.2.1 Estudo de caso – Criação de um site pessoal

Este estudo de caso tem como objetivo criar uma página de internet que trará informações sobre um grupo de amigos que se reúnem frequentemente, trazendo informações sobre festas, notícias e fotos. Como o responsável pela criação e administração do site não possui profundos conhecimentos em programação para *web*, decidiu utilizar um software de

gerenciamento de conteúdo.

Após assinar um plano de hospedagem em um *host* qualquer de internet que contém os requisitos para a execução do software, recebeu as informações necessárias para poder começar a acessar sua página. Primeiramente, este usuário faz uma cópia do sistema para o servidor (geralmente através de acesso ftp) e após isso já pode acessar o sistema..

Na primeira execução, o sistema orientará o usuário a preencher algumas informações necessárias para o funcionamento. São informações de como será executado o acesso ao banco de dados do servidor e esses dados geralmente são passados ao usuário quando este assina o plano de hospedagem juntamente à empresa que fornece essa hospedagem.

Depois de configurado a forma como o sistema acessará o banco de dados, pode-se acessar a página principal. Por padrão, o sistema já apresenta algumas informações visando auxiliar o usuário na utilização. O usuário pode então acessar a página de administração, onde informará alguns parâmetros utilizados na página. A figura 11 mostra essa tela de configurações. Nesta mesma etapa, o usuário escolhe o *template* que será utilizado no *site*.

TCCMS

Seja bem vindo Jean. Selecione a opção desejada:.

Configurar Site Configurar Seções Configurar Conteúdos Gerenciar Arquivos Gerenciar Usuários Efetuar Logoff

Configurar Site

Endereço do site:

Título do site:

META-TAG Keywords:

META-TAG Description:

Template do site:

Imagem de Logo:

Copyright:

Concluído Intranet local

Figura 11 – Página de configuração do sistema

Após configurar os parâmetros, o usuário pode então começar a publicar conteúdos e seções. As seções dizem respeito às informações que irão aparecer no menu principal do site. Elas podem ser *links* chamando outros *sites* ou então podem ser *links* para arquivos que o usuário publicou na página.

Os conteúdos são informações que são apresentadas na página principal do site. Um conteúdo deve conter uma breve descrição do acontecimento ou do anúncio efetuado e pode conter uma extensão mais detalhada sobre essa informação. Nesse caso, será apresentado uma opção informando que há uma extensão dessa informação, incluindo um *link* que abre uma página mostrando-a.

Na figura 12, pode ser observada a criação de um conteúdo para a página, enquanto que na figura 13 é apresentada a tela de criação de seções.



Figura 12 – Criação de conteúdo para o site



Figura 13 – Criação de seções para o site

O usuário também pode disponibilizar arquivos utilizando este sistema de gerenciamento de conteúdo. Para efetuar a cópia de um arquivo localizado em sua máquina e disponibilizar em seu *site*, o usuário pode acessar a opção Gerenciar Arquivos disponibilizada

no *menu*. Será exibida uma página solicitando o nome do arquivo e o destino onde ele será copiado. Preenchendo essas informações, e clicando em Enviar, o arquivo será transferido para o destino informado no servidor. Dessa maneira, o gerenciamento de arquivos no servidor fica centralizado ao sistema de gerenciamento de conteúdo, não necessitando de outros meios tradicionais de transferência, como o FTP. A principal limitação é que este processo permite o upload de apenas um arquivo por vez. Na figura 14 é apresentada a tela de gerenciamento de arquivos.



Figura 14 – Gerenciamento de arquivos no site

O resultado final apresentado no *site* do usuário surge após efetuar as configurações necessárias utilizando a página de configurações. A administração do conteúdo que será apresentado é a operação subsequente a da configuração, cujo resultado pode ser visualizado na fig 15.

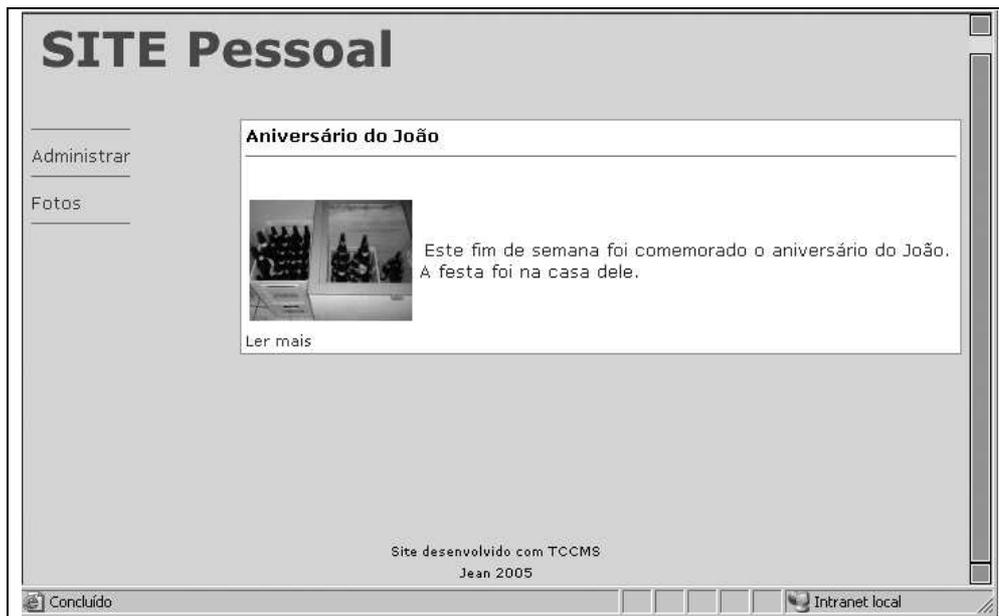


Figura 15 – Representação do site com *template* padrão

Alterando-se nas configurações, o *template* utilizado para representar o *site*, o usuário pode modificar a aparência da página, sem precisar modificar o conteúdo dela. A figura 16 apresenta uma representação do site utilizando o *template* simples.



Figura 16 – Representação do site com *template* simples

3.4 RESULTADOS E DISCUSSÃO

A utilização do sistema implementado, comparado aos outros sistemas citados nos trabalhos correlatos, mostra-se muito eficaz levando-se em conta a facilidade de utilização. Isso acontece porque os sistemas pesquisados incluem muitos outros recursos que acabam deixando sua utilização complexa.

O desenvolvimento buscou reunir o que cada um dos trabalhos correlatos possui como ponto forte, sendo essas características apresentadas na tabela 1. Além de incorporar essas características, os requisitos básicos também foram atendidos.

Tabela 1 – Características de Sistemas de gerenciamento de conteúdos

SISTEMA	CARACTERÍSTICA UTILIZADA
JetBox	<i>wizard</i>
Mambo	<i>templates</i>
E107	administração

O *wizard* foi incorporado para auxiliar o usuário na configuração inicial do acesso ao site. É através dele que o usuário fornecerá as informações necessárias para acessar o banco de dados.

A possibilidade de criação de *templates* fornece um mecanismo para que o usuário possa modificar a aparência de seu site de uma maneira simples.

Já a página de administração permite ao usuário administrador definir as informações que serão apresentadas em seu site.

O desenvolvimento do sistema conseguiu unir essas características, aprimorando assim esses trabalhos correlatos em um só sistema. Apesar da união das principais características desses sistemas, notou-se que não houve comprometimento do desempenho da aplicação.

4 CONCLUSÕES

O objetivo principal deste trabalho que era desenvolver um sistema de gerenciamento de conteúdo para *web* utilizando como linguagem de programação PHP 5 e banco de dados o MySQL foi alcançado. A utilização desta linguagem mostrou-se adequada para garantir a implementação do sistema.

São poucas as aplicações existentes que suportam amplamente essa plataforma de execução. Talvez por ter sido lançada há pouco tempo, ainda não ganhou a confiança plena dos programadores. A expectativa é que a partir de agora, a adoção do PHP 5 como linguagem de desenvolvimento para a internet, seja reconhecida como uma alternativa segura e seu uso aumente em escala gradativa, substituindo assim o PHP 4.

O uso do sistema criado permite ao usuário-administrador do *site* criar e manter uma página de internet, sem exigir grandes conhecimentos em programação. Utilizando ferramentas que auxiliam o usuário, a interface do sistema para o gerenciamento de informações é semelhante aos aplicativos utilizados diariamente por esses usuários, facilitando assim sua utilização.

Para a especificação do sistema, a utilização da ferramenta *Power Designer* facilitou a elaboração do projeto. Apesar desta ferramenta não gerar código para o PHP baseado na especificação criada, ajudou a ter uma visão geral do conceito do sistema.

Visando testar uma das novas funcionalidades do PHP 5, o sistema foi desenvolvido focando a orientação a objetos. Apesar de não ser necessário a utilização de aspectos avançados desta metodologia, a linguagem mostrou-se muito satisfatória na elaboração do sistema, atendendo completamente a necessidade.

Comparado com outros sistemas de gerenciamento de conteúdos existentes atualmente, o sistema desenvolvido apresenta algumas limitações, tais como:

- a) modularização do sistema, o que facilitaria assim a integração de novas funcionalidades através dos chamados *plugins* ;
- b) falta de funcionalidades adicionais presentes em outros sistemas de gerenciamento de conteúdo;
- c) ausência de um mecanismo que possa auxiliar usuários sem conhecimentos em programação a criar novos *templates*, tendo em vista que para isso atualmente é necessário possuir um pouco de conhecimento em programação HTML.

4.1 EXTENSÕES

Para a extensão de trabalhos futuros sugere-se:

- a) modularização através de *plugins*;
- b) gerenciamento de usuários integrados com sistemas *Lightweight Directory Access Protocol* (LDAP);
- c) elaboração de estatísticas de acesso;
- d) criação de um mecanismo para auxílio na elaboração de *templates*;
- e) desenvolvimento de um mecanismo para exportação das páginas, afim de ser utilizado em hosts que não atendam os requisitos básicos do software.

REFERÊNCIAS BIBLIOGRÁFICAS

ABITEBOUL, Serge; BUNEMAN, Peter; SUCIU, Dan. **Gerenciando dados na web**. Rio de Janeiro: Campus, 2000.

BAZON, Mihai. **HTMLArea**. [S.l.], 2003. Disponível em: <<http://www.ukesf.net/apc-aa/misc/htmlarea/>>. Acesso em: 27 mai. 2005.

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

E107.ORG. **e107.org**. [S.l.], 2004. Disponível em: <<http://e107.org/news.php/>>. Acesso em: 30 set. 2004.

HUNTER, Philip. **The management of content: universities and the electronic publishing revolution**. [S.l.], 2001. Disponível em: <<http://www.ariadne.ac.uk/issue28/cms/intro.html>>. Acesso em: 15 maio de 2005.

IBOPE. **Projeções da internet no Brasil**. Brasil, 2004. Disponível em: <http://www.ibope.com.br/imprensa/noticias_2004_internetjan_no.htm>. Acesso em: 30 set. 2004.

JETBOX. **JetBox One**. [S.l.], 2004. Disponível em: <<http://jetboxone.sourceforge.net/>>. Acesso em: 30 set. 2004.

MACROMEDIA . **RoboHelp X5**. [S.l.], 2004. Disponível em: <<http://www.macromedia.com/software/robohelp/>>. Acesso em: 24 out. 2004

MAMBO SERVER. **Mambo**. [S.l.], 2004. Disponível em: <<http://www.mamboserver.com/>>. Acesso em: 30 set. 2004.

MORATELLI, Alexandre. **Sistema de gerenciamento de conteúdo para ambiente web**. 2002. 59 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PHP Group. **PHP: hypertext preprocessor**. [S.l.], 2004. Disponível em: <<http://www.php.net/>>. Acesso em: 30 set. 2004.

TRACHTENBERG, Adam. **Why PHP 5 rocks**. [S.l.], 2004. Disponível em: <<http://www.onlamp.com/pub/a/php/2004/07/15/UpgradePHP5.html>>. Acesso em: 05 maio de 2005.

SOARES, Wallace. **Programando em PHP**. Rio de Janeiro: Érica, 2000.

APÊNDICE A – Página de administração – admin.php

```

<?
// Inicializa sessão.
session_start();
/*****
*
*                               TCCMS
*
* Arquivo      : admin.php
* Autor        : Jean Carlos Bormanieri
* Início       : 01/04/2005
* Alteração   : 18/04/2005
*
*****
* Função       : Pagina de Administração
*****/
?>
<html>
<head>
<link rel="stylesheet" type="text/css" href="./templates/padrao/admin.css">
<title>TCCMS - Página de Administração</title>
<!-- HTMLAREA-->
<script language="Javascript1.2"><!-- // load htmlarea
_editor_url = "htmlarea/"; // URL to htmlarea files
var win_ie_ver = parseFloat(navigator.appVersion.split("MSIE")[1]);
if (navigator.userAgent.indexOf('Mac') >= 0) { win_ie_ver = 0; }
if (navigator.userAgent.indexOf('Windows CE') >= 0) { win_ie_ver = 0; }
if (navigator.userAgent.indexOf('Opera') >= 0) { win_ie_ver = 0; }
if (win_ie_ver >= 5.5) {
document.write('<scr' + 'ipt src="' + _editor_url + 'editor.js"');
document.write(' language="Javascript1.2"></scr' + 'ipt>');
} else { document.write('<scr'+ 'ipt>function editor_generate() { return false;
}</scr'+ 'ipt>'); }
// --></script>
<!-- HTMLAREA-->
<SCRIPT defer>
var fx_imgwin;
var fx_opener;

function OpenFile(oOpener)
{
fx_opener = oOpener;
fx_imgwin = window.open('loadfile.php', 'selImg', 'width=640, height=480,
scrollbars=yes');
fx_imgwin.focus();
}

function onImage(u) {
fx_opener.value = u;
fx_imgwin.close();
}
</script>
</head>
<body topmargin="0" leftmargin="0">

<?
// Inclui arquivo de definição de classes
include('classes/db.php');
// Verificar se esta vindo dados de um post e recupera ele
if (isset($_HTTP_POST_VARS["lguser"])){
$login = $_HTTP_POST_VARS["lguser"];
$senha = $_HTTP_POST_VARS["pwuser"];
//Criptografar a senha
$senha = md5($senha);
//Criar uma conexão com o banco de dados
$conexao = new db;

if ($conexao->isAdmin($login,$senha)) {
// Logou com sucesso
// Vamos criar a sessão
$_SESSION["admin"] = "1";
// Recuperar o nome do usuário, baseado no login dele
//

```

```

        $_SESSION["userName"] = $conexao->getUsername($login);
        $_SESSION["codUser"] = $conexao->getCodUser($login);
    }else{
        $erro = "Erro ao efetuar logon. Tente novamente";
    }
}
// Verificar se a sessão está setada

if (isset($_SESSION["admin"])){
    // Mostrar página de administrador
    include('templates/padrao/adm00.tpl');
    /* A partir de agora, será tratado cada operação que o usuário desejar*/
    // Verificar se está setado alguma operação
    if (isset($_HTTP_GET_VARS["operation"])){
        // Recupera o código da operação
        $operacao = $_HTTP_GET_VARS["operation"];
        $mensagem = "";
        switch ($operacao)
        {
            case "1":
                include('templates/padrao/adm01.tpl');
                break;
            case "2":
                include('templates/padrao/adm02.tpl');
                break;
            case "3":
                include('templates/padrao/adm03.tpl');
                break;
            case "4":
                include('templates/padrao/adm04.tpl');
                break;
            case "5":
                include('templates/padrao/adm05.tpl');
                break;
        } //end switch
    } //end if
}else{ //Erro ao logar
    ?>
    <br>
    <table class="borda" width="250" align="center">
    <FORM METHOD=POST ACTION="admin.php">
    <tr><td align="center" colspan="2">TCCMS</td></tr>
    <tr><td align="right">Login.: </td><td align="center">
    <INPUT TYPE="text" NAME="lguser" size="20"></td></tr>
    <tr><td align="right">Senha.: </td><td align="center">
    <INPUT TYPE="password" NAME="pwuser" size="20"></td></tr>
    <tr><td align="center" colspan="2">
    <INPUT TYPE="submit" value="Administrar"></td></tr>
    </FORM>
    </table>
    <? // Mostrar erro
        if (isset($erro)){
    ?>
    <br><table class="borda" width="250" align="center">
    <tr><td align="center">
    <? print $erro; ?>
    </td></tr>
    </table>
    <?}>
    <?}>
</html>

```

APÊNDICE B – Página principal – index.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : index.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 31/05/2005
*
*****
* Função    : Mostrar a página Inicial
*****/

// Verificar se as configurações iniciais já foram efetuadas,
// senão, redirecionar para a página de instalação.
$dir_install = 'linstall.php';

if (file_exists($dir_install)) {
    header('location:'.$dir_install);
} else {
    //Inclui arquivo de definição de classes
    include('classes/db.php'); include('classes/template.php');

    //Criar uma conexão com o banco de dados para buscar informações do banco
    $banco = new db;

    //Ler as configurações do banco de dados para saber qual template será carregado
    $template_dir = 'templates/'.$banco->getConfig("templatesite");

    $template = new template;

    if (isset($_HTTP_GET_VARS["content"])){
        $template->loadTemplateContent($template_dir, $_HTTP_GET_VARS["content"]);
    }else{
        $template->loadTemplate($template_dir);
    }
}
?>

```

APÊNDICE C – Página auxiliar de instalação – install.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : install.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 13/04/2005
*
*****
* Função    : Auxiliar o usuário na configuração básica do aplicativo *
*****/
?>
<html>
<style>
td { font-family:verdana,arial;
      font-size:11px;
    }
</style>

<body>
<?
// Recuperar informações de postagem para podermos identificar em qual passo estamos.
// Se não encontrar essa informação, então mostrar página inicial.
//
if(isset($_HTTP_POST_VARS["step"])) {
    $step = $_HTTP_POST_VARS["step"];
}else{
    $step = "0";
}
//Utilizar um switch para identificar o passo em que estamos
switch ($step)
{
    //Passo 1
    case "1":
        // No passo 1, será configurado o acesso ao banco de dados
        // Inclui as variáveis de acesso ao banco de dados
        include('conf/config.php');
        // Inclui o template do passo 1
        include_once('install/step1.tpl');
        //Finaliza o passo 1
        break;
    //Passo 2
    case "2":
        // No passo 2, será configurado as informações iniciais do site
        // Inclui o template do passo 2
        include_once('install/step2.tpl');
        //Finaliza o passo 2
        break;
    //Passo 3
    case "3":
        // No passo 3, gravar as informações no banco de dados
        //Verifica se as informações vieram do assistente
        if ((isset($_HTTP_POST_VARS["lguser"])) &&
            (isset($_HTTP_POST_VARS["nmuser"]))) {

            //Recuperar os dados postados
            $lguser = $_HTTP_POST_VARS["lguser"];
            $nmuser = $_HTTP_POST_VARS["nmuser"];
            $pwuser = $_HTTP_POST_VARS["pwuser"];
            // Encriptar a senha do usuário
            $pwuser = md5($pwuser);
            // Inclui arquivo de definição de classes
            include('classes/db.php');

            // Criar um objeto de conexão com o banco de dados
            $conexao = new db;

            // Agora que possuímos uma conexão com o banco de dados,
            // vamos gravar as informações iniciais no banco

```

```

        $sql = "INSERT INTO usuario (NMUSER, LGUSER, PWUSER) VALUES
              ('".$nmuser."', '".$lguser."', '".$pwuser."')";
        $result = $conexao->execSql($sql);

        if ($result) { // O cadastro ocorreu corretamente
            // Como todos os passos do assistente foram executados
            // corretamente encerrar o assistente
            // Inclui o template do passo 3
            include_once('install/step3.tpl');

        }else{ //Erro ao inserir no banco de dados
            print "Erro ao executar a seguinte Query: ".$sql;
        }
    }else{
        print "ERRO ao ler informações de Postagem.";
    }
    //Finaliza o passo 3
    break;

//
//Por default, mostrar a tela inicial da página de configuração
default:
    //
    // Inclui o template padrao
    //
    include_once('install/step.tpl');
    break;
}

?>

</body>
</html>

```


APÊNDICE D – Listar arquivos de imagens – loading.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : loading.php
* Autor    : Jean Carlos Bormanieri
* Início   : 01/04/2005
* Alteração : 10/06/2005
*
*****
* Função    : Carregar as imagens utilizadas no editor Html
*****/

function mostraImagens() {
    // variável que define o diretório das imagens
    $dir = "./media/";
    // esse seria o "handler" do diretório
    $dh = opendir($dir);
    // loop que busca todos os arquivos até que não encontre mais nada
    while (false !== ($filename = readdir($dh))) {
        if (($filename != ".") && ($filename != "..")){
            // mostra o nome do arquivo e um link para ele - pode ser mudado para
            mostrar diretamente a imagem :)
            echo "<A
href='javascript:select_image(\"$dir$filename\")'>\".$filename.\"</a><br>";
        } //Fim If
    } // Fim While
} // Fim funcao
?>
<html>
<head>
<link rel="stylesheet" type="text/css" href="./templates/padrao/padrao.css">
</head>
<body>
<script language=javascript>
var g_sImagePath;
//internal
function select_image(sImagePath){
if (sImagePath.indexOf(".gif")>0 || sImagePath.indexOf(".jpg")>0 ||
sImagePath.indexOf(".jpeg")>0 || sImagePath.indexOf(".bmp")>0 ||
sImagePath.indexOf(".ico")>0)
{document.all.preview.src = sImagePath; }
else
{sImagePath = "http://" + document.domain + "/" +
sImagePath;
document.all.preview.src = "admin/media/zip.ico";
}
g_sImagePath = sImagePath;
document.all.previewtext.innerText = sImagePath;
}

function CheckSize(){
if (document.all.preview.width>220){ document.all.preview.width=220;}
//external
function insert_image() {
var sImg = g_sImagePath;
eval(parent).opener.onImage(sImg, 'media' , imgfrm.bdr.value,
imgfrm.algn.value, imgfrm.alt.value, 1);
val = close();
}
</script>

<table width="100%" border=0 cellpadding=0 cellspacing=0>
<tr><td align="center">TCCMS HtmlEditor</td></tr>
</table>
<table width="100%" border=0 cellpadding=0 cellspacing=0 height="90%"><tr>
<td align=center valign=middle>
<table width=90% height=70% style="border: 1px black solid; background: white;">
<tr>
<td align=center valign=middle>

```


APÊNDICE E – Efetuar logoff do sistema – logoff.php

```

<?
// Inicializa sessão.
session_start();
/*****
*
*                               TCCMS
*
* Arquivo   : logoff.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 09/05/2005
*
*****
* Função    : Finalizar a sessão do usuário
*****/
?>
<html>
<head>
<link rel="stylesheet" type="text/css" href="./templates/padrao/admin.css">
</head>
<body topmargin="0" leftmargin="0">

    <? // verificar se a sessão está aberta
        if (isset($_SESSION["admin"])){

                // Finalizar a sessão do usuário
                session_destroy();

            ?>
<table border="0" width="100%" cellspacing="0" cellpadding="0" bgcolor="#6384A5">
<tr><td><IMG SRC="templates/padrao/logotccms.gif" BORDER="0" ALT=""></td></tr>
</table>
<br>
<table border="0" cellspacing="5" cellpadding="10" class="borda" align="center">
<tr><td align="center">Sua sessão foi finalizada corretamente.</td>
</tr>
<tr>
<td align="center">
Clique <A HREF="index.php">aqui</A> para voltar a página principal.</td>
</tr>
</table>
<? }// fim if ?>
</body>
</html>

```

APÊNDICE F – Classe Template – template.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo      : template.php
* Autor       : Jean Carlos Bormanieri
* Início      : 01/04/2005
* Alteração   : 31/05/2005
*
*****
* Função      : Classe de Template
*****/

class template {

    function loadtemplate($template) {

        //Carrega o arquivo principal do Template
        $main = implode("", @file($template."/main.tpl"));
        $this->replaceConfig($main, $template);
        $this->replaceSecao($main, $template);
        $this->replaceConteudo($main, $template);
        print $main;
    }

    function loadtemplateContent($template, $codconteudo) {

        //Carrega o arquivo principal do Template
        $main = implode("", @file($template."/main.tpl"));
        $this->replaceConfig($main, $template);
        $this->replaceSecao($main, $template);
        $this->loadConteudo($main, $template, $codconteudo);
        print $main;
    }

    function replaceConteudo(&$pagina, $template){
        //Cria novo conteudo
        $novoConteudo = new conteudo;

        //Pegar lista de conteudos
        $listacont = $novoConteudo->getListaConteudo("all");

        //Criar lista de conteudos
        $conteudos = "";

        while ($row = mysql_fetch_array($listacont)){
            //Carregar arquivo de template
            $arqcont = implode("", @file($template."/conteudo.tpl"));

            //Item que será adicionado
            $item = "";
            $item = $item."<!--Inicio Conteudo--><table border='0' height='175'
width='100%' class='borda' ><tr valign='top'><td class='tituloconteudo' height='10'>";

            //Carrega o titulo do Conteudo
            $item = $item . $row['TITCONT']."<br><HR align=left width='100%' SIZE=1
shade></td></tr>";

            //Carregar a Descrição do Conteudo
            $item = $item."<tr><td><br>".$row['DSTCONT'];

            if ($row['COPCONT'] != ""){
                //Colocar um link para o conteúdo
                $item = $item."</td></tr><tr><td><a
href='index.php?content=".$row['CODCONT']."'>Ler mais</a>";
            }

            //Finaliza o item
            $item = $item."</td></tr></table><!--Fim Conteudo-->";

            $item = "<tr valign='top'><td width='100%'>".$item."</td>";
        }
    }
}

```

```

        //Inclui o item
        $arqcont = str_replace("{ITEMCONTEUDO}", $item, $arqcont);

        //Concatenar o resultado
        $conteudos = $conteudos.$arqcont;
    }

    //Agora, alterar no template Main
    $pagina = str_replace("{CONTEUDO}", $conteudos, $pagina);
}

function replaceSecao(&$pagina, $template){
    //Cria uma seção
    $novaSecao = new secao;

    //Incluir as seções

    //Pegar lista de secões
    $listasec = $novaSecao->getListaSecao("*");

    //Criar lista de seção
    $secoes = "";

    while ($row = mysql_fetch_array($listasec)){
        //Carregar arquivo de template
        $arqsecoes = implode("", @file($template."/secao.tpl"));
        //Verifica se é um link
        if ($row['LNKEXTERNO'] != ""){
            $arqsecoes = str_replace("{ITEMSECAO}", "<a
href=\"".$row['LNKEXTERNO'].\">".$row['DSCSECAO'].\"</a>", $arqsecoes);
        }else{
            $arqsecoes =
str_replace("{ITEMSECAO}", $row['DSCSECAO'], $arqsecoes);
        }
        $secoes = $secoes.$arqsecoes;
    }
    //Agora, alterar no template Main
    $pagina = str_replace("{SECAO}", $secoes, $pagina);
}

function replaceConfig(&$pagina, $template){
    //Cria uma conexão com o banco de dados
    $banco = new db;

    //Cria um array que conterá as palavras a serem substituídas
    $dados = array('{TITULOSITE}',
                    '{METAKEYWORD}',
                    '{METADESCRIPTION}',
                    '{COPYRIGHT}'
                );

    //Cria um array que conterá os campos que farão a substituição
    $configuracoes = array('titsite',
                            'metkey',
                            'metdesc',
                            'cpright'
                );

    //Descobre o tamanho do array
    $total = sizeof($dados);

    //Substitui na página
    for($i=0; $i <$total; $i++){
        $pagina = str_replace($dados[$i], $banco-
>getConfig($configuracoes[$i], $pagina);
    }

    //Substitui o logo da página, se houver
    if ($banco->getConfig("imglogo")) {
        $temp = "<img src='".$banco->getConfig("imglogo")."' border='0'>";
        $pagina = str_replace('{LOGO}', $temp, $pagina);
    } //End if
}

function loadConteudo(&$pagina, $template, $codigo){
    //Cria uma conexão com o banco de dados
    $novoConteudo = new conteudo;

```

```

//Pegar lista de conteudos
$listacont = $novoConteudo->getListaConteudo($codigo);
//Criar lista de conteudos
$conteudos = "";
while ($row = mysql_fetch_array($listacont)){
    //Carregar arquivo de template
    $arqcont = implode("", @file($template."/conteudo.tpl"));

    //Item que será adicionado
    $item = "";
    $item = $item."<!--Inicio Conteudo--><table border='0' width='100%'
class='borda' ><tr valign='top'><td class='tituloconteudo' height='10'>";
    //Carrega o titulo do Conteudo
    $item = $item . $row['TITCONT']. "<br><HR align=left width='100%' SIZE=1
shade></td></tr>";

    //Carregar a Descrição do Conteudo
    $item = $item."<tr><td><br>". $row['DSTCONT'];
    $item = $item."</td></tr><tr><td>". $row['COPCONT'];
    //Finaliza o item
    $item = $item."</td></tr></table><!--Fim Conteudo-->";
    $item = "<tr valign='top'><td width='50%'>". $item."</td></tr>";
    //Inclui o item
    $arqcont = str_replace("{ITEMCONTEUDO}", $item, $arqcont);
    //Concatenar o resultado
    $conteudos = $conteudos.$arqcont;
}

//Agora, alterar no template Main
$pagina = str_replace("{CONTEUDO}", $conteudos, $pagina);
}
?>

```

APÊNDICE G – Classe conteudo – conteudo.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : conteudo.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 14/06/2005
*
*****
* Função    : Classe de Conteudo
*****/

class conteudo {

    function getListaConteudo($codigo){

        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        if ($codigo == "0"){
            $sql = "SELECT * FROM conteudo";
        }elseif ($codigo == "all"){
            $sql = "SELECT * FROM conteudo WHERE viscont = 1 ORDER BY
codcont DESC";
        }else{
            $sql = "SELECT * FROM conteudo WHERE codcont = ".$codigo;
        }

        $resultado = mysql_query($sql);
        return $resultado;
    }

    function setConteudo($iduser, $titcont, $dstcont, $copcont, $viscont){
        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        $sql = "INSERT INTO conteudo VALUES
(NULL, ".$iduser.", ".$titcont.", ".$dstcont.", ".$copcont.", ".$viscont.")";

        //print $sql;
        $resultado = $banco->execSql($sql);
        return $resultado;
    }

    function updateConteudo($codcont, $iduser, $titcont, $dstcont, $copcont, $viscont){
        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        $sql = "UPDATE conteudo SET iduser=".$iduser." , titcont='".$titcont." ,
dstcont='".$dstcont." , copcont='".$copcont." , viscont='".$viscont." WHERE
codcont=".$codcont;

        //print $sql;
        $resultado = $banco->execSql($sql);
        return $resultado;
    }
}

```

```
function deleteConteudo($codigo){
    $banco = new db;

    // Abre o banco de dados para conexão. Seleciona o banco de dados
    $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

    //Monta a string SQL
    $sql      = "DELETE FROM conteudo WHERE codcont = ".$codigo."";

    $resultado = $banco->execSql($sql);
    return $resultado;
}
?>
```

APÊNDICE H – Classe secao – secao.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : secao.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 31/05/2005
*
*****
* Função    : Classe de Secao
*****/

class secao {

    function setSecao($titsecao, $vissecao, $lnksecao){
        $banco = new db;
        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");
        //Monta a string SQL
        $sql = "SELECT * FROM secoes WHERE dscsecao = '". $titsecao. "'";
        $resultado = mysql_query($sql) or die(" " . mysql_error());

        if (mysql_num_rows($resultado) > 0) { //Já existe o atributo, fazer um update
            $sql = "UPDATE secoes SET visivel = ".$vissecao.",
lnkexterno='". $lnksecao.'" WHERE dscsecao = '". $titsecao. "'";
        }else{
            $sql = "INSERT INTO secoes VALUES
(NULL, '". $titsecao. "', ". $vissecao. ", '". $lnksecao. "')";
        }
        $resultado = $banco->execSql($sql);
        return $resultado;
    }

    function deleteSecao($titsecao){
        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        $sql = "DELETE FROM secoes WHERE dscsecao = '". $titsecao. "'";

        $resultado = $banco->execSql($sql);
        return $resultado;
    }

    function getListaSecao($codigo){
        $banco = new db;
        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");
        //Monta a string SQL
        if ($codigo == "0"){
            $sql = "SELECT * FROM secoes";
        }elseif ($codigo == "**"){
            $sql = "SELECT * FROM secoes WHERE visivel = 1 ORDER BY
dscsecao";
        }else{
            $sql = "SELECT * FROM secoes WHERE codsecao = ".$codigo;
        }
        $resultado = mysql_query($sql);
        return $resultado;
    }
}
?>

```

APÊNDICE I – Classe usuario – usuario.php

```

<?
/*****
*
*                               TCCMS
*
* Arquivo   : usuario.php
* Autor     : Jean Carlos Bormanieri
* Início    : 01/04/2005
* Alteração : 31/05/2005
*
*****
* Função    : Classe de Usuario
*****/

class usuario {

    function getListaUsuarios($codigo){
        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        if ($codigo == "0"){
            $sql = "SELECT * FROM usuario";
        }elseif ($codigo == "all"){
            $sql = "SELECT * FROM usuario ORDER BY LGUSER";
        }else{
            $sql = "SELECT * FROM usuario WHERE lguser = '". $codigo. "'";
        }

        $resultado = mysql_query($sql);
        return $resultado;
    }

    function isAdmin($usuario, $senha){
        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        $sql = "SELECT * FROM usuario WHERE lguser = '". $usuario. "' AND pwuser =
'". $senha. "'";
        //print $sql;

        $resultado = mysql_query($sql) or die(" " . mysql_error());
        $resultado = mysql_num_rows($resultado);
        return $resultado;
    }

    function getUsername($login){
        $banco = new db;

        // Abre o banco de dados para conexão. Seleciona o banco de dados
        $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

        //Monta a string SQL
        $sql = "SELECT * FROM usuario WHERE lguser = '". $login. "'";
        //print $sql;

        $resultado = mysql_query($sql) or die(" " . mysql_error());
        $resultado = mysql_fetch_array($resultado);
        return $resultado["NMUSER"];
    }
}

```

```

function getCodUser($login){
    $banco = new db;

    // Abre o banco de dados para conexão. Seleciona o banco de dados
    $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

    //Monta a string SQL
    $sql = "SELECT * FROM usuario WHERE lguser = '". $login. "'";
    //print $sql;

    $resultado = mysql_query($sql) or die(" " . mysql_error());
    $resultado = mysql_fetch_array($resultado);
    return $resultado["IDUSER"];
}

function getUserbyCodigo($codigo){
    $banco = new db;

    // Abre o banco de dados para conexão. Seleciona o banco de dados
    $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

    //Monta a string SQL
    $sql = "SELECT * FROM usuario WHERE iduser = '". $codigo. "'";
    //print $sql;

    $resultado = mysql_query($sql) or die(" " . mysql_error());
    $resultado = mysql_fetch_array($resultado);
    return $resultado["NMUSER"];
}

function setUsuario($lguser, $nmuser, $pwuser){
    $banco = new db;

    // Abre o banco de dados para conexão. Seleciona o banco de dados
    $con = mysql_connect($banco->server,$banco->user,$banco->password) or
die("CONNECTION ERROR!!!");
        mysql_select_db($banco->database) or die("DATABASE ERROR!!!");

    //Monta a string SQL
    $sql = "INSERT INTO usuario VALUES
(NULL, '$nmuser.', '$lguser.', '$pwuser.')";

    //print $sql;
    $resultado = $banco->execSql($sql);
    return $resultado;
}
}
?>

```