

**Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática**

**UM ESTUDO COMPARATIVO DE SISTEMAS DE GESTÃO DE
CONTEÚDO PARA A MANUTENÇÃO DE SÍTIOS WEB EM
UNIVERSIDADES PÚBLICAS BRASILEIRAS**

Leandro Koiti Sato

Maringá - Paraná

Brasil

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática

**UM ESTUDO COMPARATIVO DE SISTEMAS DE GESTÃO DE
CONTEÚDO PARA A MANUTENÇÃO DE SÍTIOS WEB EM
UNIVERSIDADES PÚBLICAS BRASILEIRAS**

Leandro Koiti Sato

Monografia apresentada como requisito parcial à obtenção do grau de Especialista pelo curso de Pós-Graduação em Desenvolvimento de Sistemas para Web da Universidade Estadual de Maringá.

Orientador: *Prof. Dr. Dante Alves Medeiros Filho*

**Maringá - Paraná
2008**

Leandro Koiti Sato

**UM ESTUDO COMPARATIVO DE SISTEMAS DE GESTÃO DE
CONTEÚDO PARA A MANUTENÇÃO DE SÍTIOS WEB EM
UNIVERSIDADES PÚBLICAS BRASILEIRAS**

Este exemplar corresponde à redação final da monografia aprovada como requisito parcial para obtenção do grau de Especialista pelo curso de Pós-Graduação em Desenvolvimento de Sistemas para Web da Universidade Estadual de Maringá, pela comissão formada pelos professores:

Orientador: Prof. Dante Alves Medeiros Filho
Departamento de Informática, CTC, DIN

Prof. José Roberto Vasconcelo
Departamento de Informática, CTC, DIN

Prof. Paulo Cesar Gonçalves
Departamento de Informática, CTC, DIN

Maringá, Novembro de 2008

Universidade Estadual de Maringá
Departamento de Informática
Av. Colombo 5790, Maringá-PR
CEP 87020-900
Tel: (44) 32614324 Fax: (44) 3263587

Agradecimentos

Ao professor Dante Alves Medeiros Filho, que me ajudou e me orientou ao longo do desenvolvimento deste trabalho.

Aos meus pais e familiares, pelo amor, apoio e incentivo aos meus estudos. Sem eles, a realização deste trabalho não seria possível.

A todos os professores do curso de especialização, que colaboraram com a minha formação.

À toda a equipe de servidores da UTFPR, em especial ao professor Paulo Apelles Camboim de Oliveira, que autorizou o horário diferenciado para que eu pudesse retornar a Maringá as sexta-feiras e terminasse de assistir às aulas do curso de especialização.

Enfim, a todos aqueles que contribuíram diretamente, ou indiretamente à realização deste trabalho.

Resumo

No começo da *World Wide Web*, os sítios eram simples, pequenos e projetados informalmente. A tarefa de desenvolvimento e manutenção do sítio era delegada a uma única pessoa chamada *webmaster*. As páginas *web* eram estáticas, escritas em puro HTML e não havia maneira de separar o seu aspecto visual do seu conteúdo, portanto este profissional era responsável por fazer todas estas tarefas. Conforme as páginas cresciam, tornando sítios grandes e mudando rapidamente, a necessidade por fortes ferramentas de gestão de conteúdos se tornou premente. Neste contexto surgem os sistemas de gestão de conteúdos (CMS), que permitem um site crescer e mudar rapidamente enquanto mantém sua alta qualidade. Como existem várias soluções de CMS, o presente trabalho realiza um estudo visando subsidiar a escolha desta tecnologia quando do desenvolvimento e da manutenção de sítios *web* para universidades públicas brasileiras. Para isto, é usado como estudo de caso a Universidade Federal do Paraná (UTFPR), campus Francisco Beltrão.

Abstract

In the beginning of the World Wide Web, sites were simple, small and informally designed. The task of designing and maintaining the website was delegated to a single person called the webmaster. The webpages were static, written in plain HTML and there was no way to separate the visual design from the content, hence this professional was responsible to do all those tasks. As the pages grew up turning into large and rapidly changing sites, the need for strong management tools became pressing. In this context emerged the Content Management Systems (CMS), which allows a site grow and change quickly while maintaining high quality. As there are many CMS solutions, the current work perform a study aiming support the choice of this technology in the development and maintaining of Brazilian public universities. In order to do this, it was used as case study the Federal University of Technology – Parana (UTFPR), Francisco Beltrao Campus.

Índice

1 INTRODUÇÃO	1
<hr/>	
1.1 MOTIVAÇÃO	1
1.2 OBJETIVOS GERAIS	2
1.3 OBJETIVOS ESPECÍFICOS	2
1.4 METODOLOGIA	2
1.5 APRESENTAÇÃO	3
2 FUNDAMENTAÇÃO TEÓRICA	4
<hr/>	
2.1 INTRODUÇÃO	4
2.2 DADOS, INFORMAÇÃO E CONTEÚDO	4
2.3 METADADOS	5
2.4 <i>WORKFLOW</i>	8
2.5 GESTÃO DE CONTEÚDO	9
2.6 SISTEMA DE GESTÃO DE CONTEÚDO	9
2.7 PADRÕES BRASIL E-GOV	12
2.8 PARÂMETROS PARA ANALISAR UM CMS	13
2.9 CONSIDERAÇÕES FINAIS	15
3 MATERIAIS E MÉTODOS	16
<hr/>	
3.1 INTRODUÇÃO	16
3.2 CRITÉRIOS	16
3.3 VISÃO GERAL DO PROJETO DE CMS	17
3.4 PESQUISA SOBRE CMS DISPONÍVEIS	18
3.5 CORTE NA LISTA	19
3.6 COMPARAÇÃO ENTRE OS CMS	21
3.6.1 CRITÉRIOS TÉCNICOS	21
3.6.2 UTILIZAÇÃO DE PADRÕES WEB, E-GOV E ACESSIBILIDADE	22
3.6.3 <i>WORKFLOW</i>	22
3.6.4 INTERNACIONALIZAÇÃO E LOCALIZAÇÃO	22
3.6.5 FACILIDADE DE USO	22
3.6.6 PLUG-INS	23
3.6.7 SEGURANÇA	23
3.6.8 DOCUMENTAÇÃO E SUPORTE	23
4 ANÁLISE DOS CMS	24
<hr/>	
4.1 INTRODUÇÃO	24
4.2 DRUPAL	24
4.2.1 CARACTERÍSTICAS TÉCNICAS	24
4.2.2 PADRÕES WEB E ACESSIBILIDADE	25
4.2.3 <i>WORKFLOW</i>	25
4.2.4 INTERNACIONALIZAÇÃO E LOCALIZAÇÃO	26
4.2.5 FACILIDADE DE USO	26
4.2.6 PLUGINS	26

4.2.7	SEGURANÇA	27
4.2.8	DOCUMENTAÇÃO E SUPORTE	28
4.3	JOOMLA	29
4.3.1	CARACTERÍSTICAS TÉCNICAS	29
4.3.2	PADRÕES WEB E ACESSIBILIDADE	29
4.3.3	WORKFLOW	30
4.3.4	INTERNACIONALIZAÇÃO E LOCALIZAÇÃO	30
4.3.5	FACILIDADE DE USO	30
4.3.6	PLUGINS	31
4.3.7	SEGURANÇA	31
4.3.8	DOCUMENTAÇÃO E SUPORTE	33
4.4	PLONE	34
4.4.1	CARACTERÍSTICAS TÉCNICAS	34
4.4.2	PADRÕES WEB E ACESSIBILIDADE	34
4.4.3	WORKFLOW	35
4.4.4	INTERNACIONALIZAÇÃO E LOCALIZAÇÃO	35
4.4.5	FACILIDADE DE USO	35
4.4.6	PLUGINS	36
4.4.7	SEGURANÇA	37
4.4.8	DOCUMENTAÇÃO E SUPORTE	38
5 COMPARAÇÃO DOS CMS		39
5.1	INTRODUÇÃO	39
5.2	CARACTERÍSTICAS TÉCNICAS	39
5.3	PADRÕES WEB E ACESSIBILIDADE	39
5.4	WORKFLOW	40
5.5	INTERNACIONALIZAÇÃO E LOCALIZAÇÃO	41
5.6	FACILIDADE DE USO	41
5.7	PLUGINS	42
5.8	SEGURANÇA	43
5.9	DOCUMENTAÇÃO E SUPORTE	43
5.10	DECISÃO	44
6 CONCLUSÃO		45
REFERÊNCIAS BIBLIOGRÁFICAS		46
APÊNDICE A		A1
GLOSSÁRIO		A1
APÊNDICE B		A2
LISTA DE LIVROS PUBLICADAS SOBRE OS CMS		A2

Lista de Ilustrações

Figura 3.1 - Volume de buscas com os termos “mambo” e “joomla” (Google, 2008a).....	21
Figura 4.1 - Fragmento de código do sítio drupal.org	25
Figura 4.2 - Gravidade dos problemas de segurança do Drupal (Secunia, 2008).....	27
Figura 4.3 - Onde as falhas do Drupal poderiam ser exploradas (Secunia, 2008).....	28
Figura 4.4 - Impacto das falhas de segurança do Drupal (Secunia, 2008).	28
Figura 4.5 - Soluções para os problemas de segurança do Joomla.....	32
Figura 4.6 - Gravidade dos problemas de segurança do Joomla.	32
Figura 4.7 - Onde as falhas do Joomla poderiam ser exploradas	33
Figura 4.8 - Impacto das falhas de segurança do Joomla	33
Figura 4.9 - Gravidade dos problemas de segurança do Plone.....	37
Figura 4.10 - Onde as falhas do Plone poderiam ser exploradas.....	37
Figura 4.11 - Impacto das falhas de segurança do Plone.....	38

Lista de Quadros

Quadro 3.1 - Classificação dos CMS de acordo com o PageRank e quantidade de links ...	20
Quadro 4.1 - Detalhes técnicos do Drupal	24
Quadro 4.2 - Detalhes técnicos do Joomla	29
Quadro 4.3 - Detalhes técnicos do Plone	34
Quadro 5.1 – Comparação dos critérios técnicos	39
Quadro 5.2 - Comparação dos padrões <i>Web</i> e <i>Acessibilidade</i>	40
Quadro 5.3 - Comparação dos recursos de <i>Workflow</i>	41
Quadro 5.4 - Comparação da internacionalização e localização	41
Quadro 5.5 - Comparação da facilidade de uso	42
Quadro 5.6 - Comparação do uso de <i>plug-ins</i>	42
Quadro 5.7 - Comparação da segurança	43
Quadro 5.8 - Comparação da documentação e suporte	43

Lista de Siglas

ACL – Access Control List

ANSI – American National Standards Institute

CM – Content Management

CMS – Content Management System

CSS – Cascading Style Sheets

DCMI – Dublin Core Metadata Initiative

FAQ – Frequent Asked Questions

FTP – File Transfer Protocol

IETF – Internet Engineering Task Force

HTML – HyperText Markup Language

IRC – Internet Relay Chat

ISO – International Organization for Standardization

RFC – Request for Comments

RFP – Request for Proposal

UTFPR – Universidade Tecnológica Federal do Paraná

W3C – World Wide Web Consortium

WAI – Web Accessibility Initiative

WCAG – Web Content Accessibility Guidelines

WYSIWYG – What You See is What You Get

XHTML – eXtensible Hypertext Markup Language

XML – eXtensible Markup Language

ZODB – Zope Object Data Base

Capítulo 1

Introdução

1.1 Motivação

No início da *World Wide Web*, na primeira metade dos anos 90, as páginas eram extremamente simples, continham apenas textos com informações essenciais, quase não havia uso de imagens, devido às limitações tecnológicas da época. O acesso à rede era limitado a um número pequeno de pessoas, a maioria dos sítios era criada por estudantes e pesquisadores de universidades que tinham necessidade de divulgar informações on-line.

Com o passar do tempo, após a abertura da internet para o uso comercial e a redução do custo para se estar on-line, o acesso a *web* deixou de pertencer apenas ao mundo acadêmico e passou a estar cada vez mais presente nas empresas.

Na segunda metade dos anos 90, as corporações passaram a construir os seus próprios sítios. A tarefa da confecção e manutenção dessas páginas era confiada a especialistas, denominados *webmasters*. As páginas eram estáticas, escritas em HTML (*HyperText Markup Language*), e não havia como separar o código, a apresentação visual e o conteúdo, portanto, cabia a este único profissional o trabalho de realizar todas essas tarefas. O esforço para adicionar ou atualizar o conteúdo era grande, e crescia conforme o sítio recebia mais informações e se tornava mais complexo. Mesmo em casos simples, como a adição de um link a um menu, era necessário abrir todos os documentos HTML do sítio para a alteração manual.

O tempo para a atualização do sítio era alto, e, ao mesmo tempo, a demanda para se adicionar mais informações ao sítio aumentava. As informações não vinham mais apenas de uma única fonte, membros de diversas divisões da organização necessitavam adicionar informações dos mais diferentes tipos e nos mais diferentes formatos. Com a evolução da tecnologia, o conteúdo deixou de ser apenas textual e passou a ser distribuído como documentos, gráficos, imagens, áudio, vídeo.

A gerencia do sítio estava centrada demais no *webmaster*. As pessoas precisavam dos seus conteúdos publicados no sítio, mas dependiam dele para colocá-los no ar, pois não possuíam conhecimento técnico para isso e nem era de suas responsabilidades aprenderem.

É neste contexto é que surge a necessidade de uma ferramenta para gerenciar o conteúdo de forma ágil, onde usuários possam adicionar informações ao sítio, sem precisarem ter conhecimentos técnicos de linguagens complexas, e assim, descentralizar o trabalho da manutenção do sítio. Tal ferramenta é chamada *Content Management System* (CMS) ou Sistema de Gestão de Conteúdo.

1.2 Objetivos gerais

O objetivo deste trabalho é a escolha e adoção de um CMS de código livre como solução para gerenciar o conteúdo em sítios de universidades públicas na forma de um portal disponível na *web*. Para isso, tomou-se como estudo de casos o campus Francisco Beltrão da Universidade Tecnológica Federal do Paraná (UTFPR).

1.3 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Estudar sobre gestão de conteúdo e sistemas de gestão de conteúdo;
- Estudar e comparar algumas das principais ferramentas CMS de código aberto;
- Analisar os requisitos básicos para a criação do sítio no estudo de casos: UTFPR, campus Francisco Beltrão;
- Escolher um CMS para implantação do sítio no estudo de casos: UTFPR, campus Francisco Beltrão.

1.4 Metodologia

Para a realização deste trabalho foi pesquisado os principais CMS de código livre disponíveis e a partir desta lista, foram escolhidos três deles para uma comparação detalhada. Os critérios utilizados nesta comparação foram selecionados de acordo com a análise dos principais requisitos do estudo de casos. Através desta comparação, selecionou-se um dos CMS para a adoção no estudo de casos. Uma descrição mais detalhada dos critérios utilizados na escolha e comparação do CMS pode ser encontrada no Capítulo 3.

1.5 Apresentação

O presente trabalho está disposto em seis capítulos.

O segundo capítulo contém a fundamentação teórica que apresenta os conceitos básicos para o entendimento do trabalho.

O terceiro capítulo apresenta a metodologia utilizada para se comparar e escolher o CMS.

No quarto capítulo é apresentado cada um dos CMS individualmente, mostrando suas características principais.

No quinto capítulo é feito um paralelo entre os CMS, comparando os recursos apresentados no capítulo anterior com o auxílio de quadros comparativos. Neste capítulo é feita a escolha do melhor CMS indicado para o estudo de caso.

No sexto capítulo é feita a análise, a conclusão e as considerações finais do trabalho.

Capítulo 2

Fundamentação Teórica

2.1 Introdução

Neste capítulo são apresentados alguns conceitos básicos para o entendimento de sistemas de gestão de conteúdo. Para se definir o vocabulário básico utilizado neste trabalho, é feita uma breve definição do que são dados, informação e conteúdo, estabelecendo as principais diferenças entre eles. Em seguida, fala-se sobre metadados e o seu papel na gestão de conteúdos. A seguir, é apresentado o conceito de *workflow*, gestão de conteúdo, padrões para desenvolvimento de sítios governamentais, sistema de gestão de conteúdos e parâmetros para análise de um sistema de gestão de conteúdos.

2.2 Dados, Informação e Conteúdo

Segundo Oliveira (1997), dado é qualquer elemento identificado em sua forma bruta que por si só não conduz uma compreensão de determinado fato ou situação.

Dados são definidos como uma série de observações, medidas ou fatos na forma de números, palavras, sons e/ou imagens. Os dados não têm significado próprio, mas fornecem a matéria prima a partir da qual é produzida a informação. (Boisot, 1998).

Boisot (1998), diz que a informação é definida como os dados que foram organizados de uma forma significativa. Mcgee e Prusak (1994), complementam dizendo que a informação é o resultado de dados coletados e organizados, dotados de significado e inseridos em um contexto.

Informação é o dado trabalhado que permite ao executivo tomar decisões (Oliveira, 1997).

Segundo a *ContentWatch organization*, citado por Boiko (2005), informação em sua forma bruta torna-se conteúdo quando lhe é dada uma forma usável pretendida para um ou mais propósitos. Cada vez mais, o valor do conteúdo é baseado na combinação de sua forma primária usável, junto com a sua aplicação, acessibilidade, uso, utilidade, reconhecimento de marca e unicidade.

Os computadores foram construídos para processar dados. Dados consistem em pequenos fragmentos de informações computacionais – números, palavras, imagens, sons – que têm muito do seu significado humano retirado deles. Hoje, as pessoas usam os computadores para processar conteúdo. Assim como os dados, o conteúdo também é informação, mas retém o seu significado humano e seu contexto. (Boiko, 2005).

Segundo Boiko (2005), os usuários querem que os computadores tratem conteúdos, entretanto, este é rico em contexto e significado. Usar a tecnologia que trata dados para gerenciar conteúdos não é uma tarefa fácil. Ao fazer a informação se parecer muito com dados, ela se torna mecânica, desinteressante e difícil para os consumidores entenderem. Ao fazer a informação muito rica, carregada de contexto, não é possível fazer com que um computador automatize o seu gerenciamento.

A solução é envolver a informação em um container de dados (conhecido como metadados) de modo que o computador gerencie estes dados, deixando de lado as informações.

De acordo com Boiko (2005), conteúdo é informação que é rotulada com dados de maneira que um computador possa organizar e sistematizar sua coleção, gerenciamento e publicação. Tal como um sistema, um sistema de gestão de conteúdos tem sucesso se ele pode aplicar as metodologias de dados sem remover as vantagens e o significado da informação no meio do caminho.

2.3 Metadados

A definição comumente utilizada para metadados ou metainformação é: “dados sobre dados”. O termo se refere a qualquer dado usado para ajudar a identificação, descrição e localização de recursos eletrônicos em rede. Existem vários formatos diferentes de metadados, alguns bem simples em sua descrição, outras bem complexas e ricas. (Ifla, 2005)

Um registro de metadado consiste em um conjunto de atributos, ou elementos, necessários para descrever o recurso em questão. Por exemplo, um sistema de metadados comum em bibliotecas – o catálogo da biblioteca – contém um conjunto de registro de metadados com elementos que descrevem um livro ou um outro item da biblioteca: autor, título, data de

criação ou publicação, assunto abordado e o número de chamada especificando a localização do item na estante (Dublin Core Metadata Initiative, 2008).

Metadados são importantes, pois fornecem o contexto para os dados. Sem os metadados fornecendo o contexto, os dados da empresa não podem ser entendidos corretamente (Inmon, O'neil *et al.*, 2008).

Metadados fazem o contexto e o significado da informação explícitos o suficiente para que um computador possa manipulá-los (Boiko, 2005).

A *Dublin Core Metadata Initiative* (DCMI) é uma organização dedicada a promover a adoção de padrões de interoperabilidade de metadados e desenvolver vocabulários de metadados especializados para descrever recursos que possibilitem sistemas de descoberta de informações mais inteligentes (Dublin Core Metadata Initiative, 2008).

O padrão Dublin Core, descrito pelos padrões ISO 15836, ANSI/NISO Z39.85 e IETF RFC 5013, inclui dois níveis: o simples e o qualificado. O Dublin Core simples possui um vocabulário de quinze propriedades para o uso na descrição de recursos. (Dublin Core Metadata Initiative, 2008):

1. **Título (*Title*):** O nome dado a um recurso. Tipicamente, um Título será o nome pelo qual o recurso será formalmente conhecido.
2. **Assunto (*Subject*):** Tipicamente, o Assunto será expresso como palavras chaves, frases chaves ou códigos de classificação que descrevem o tópico do recurso.
3. **Descrição (*Description*):** Um relato sobre conteúdo do recurso. Descrição pode incluir, mas não se limita a: um resumo, índice, referência a uma representação gráfica do conteúdo ou um relato a texto livre do conteúdo.
4. **Tipo (*Type*):** A natureza ou gênero do conteúdo do recurso. Tipos incluem termos descrevendo categorias genéricas, funções, gêneros, ou níveis de agregação para o conteúdo.
5. **Fonte (*Source*):** Uma referência a um recurso do qual o presente recurso é derivado. O presente recurso pode ser derivado de uma fonte na sua totalidade ou apenas em parte.

6. **Relação (*Relation*):** Uma referência a um recurso relacionado.
7. **Abrangência (*Coverage*):** A extensão ou alcance do recurso. Abrangência inclui tipicamente uma localização espacial (o nome de um lugar ou coordenadas geográficas), um período no tempo (a sua designação, data, ou intervalo de tempo), ou jurisdição (o nome de uma entidade administrativa).
8. **Criador (*Creator*):** A entidade responsável em primeira instância pela existência do recurso.
9. **Editor (*Publisher*):** Uma entidade responsável por tornar o recurso acessível. Exemplos de um Editor incluem uma pessoa, uma organização, um serviço. Tipicamente, o nome de um Editor deve ser usado para indicar a entidade.
10. **Contribuinte (*Contributor*):** Uma entidade responsável por qualquer contribuição para o conteúdo do recurso. Exemplos de Contribuinte incluem uma pessoa, ou serviço. Tipicamente, o nome de um Contribuinte deve ser usado para indicar a entidade.
11. **Direitos (*Rights*):** Informação de direitos sobre o recurso ou relativos ao mesmo. Tipicamente, este elemento deverá conter uma declaração de gestão de direitos sobre o recurso, ou uma referência a um serviço que fornecerá essa informação. Tal poderá compreender informação sobre direitos de propriedade intelectual, direitos de autor, ou outros. A ausência deste elemento não permite formular qualquer hipótese válida sobre quaisquer direitos que possam incidir sobre o recurso
12. **Data (*Date*):** Uma data associada a um evento do ciclo de vida do recurso. Tipicamente, uma Data deve ser associada à criação ou disponibilidade do recurso.
13. **Formato (*Format*):** A manifestação física ou digital do recurso. Tipicamente, o Formato deve incluir o tipo de meio do recurso, ou as suas dimensões. Este elemento deve ser usado para determinar as aplicações informáticas ou qualquer tipo de equipamento necessário para reproduzir ou operar com o recurso. Exemplos de dimensões incluem tamanho e duração.
14. **Identificador (*Identifier*):** Uma referência não ambígua ao recurso, definida em

um determinado contexto.

15. **Idioma (*Language*):** O idioma do conteúdo intelectual do recurso.

Cada elemento é opcional e pode ser repetido. O padrão não define nenhuma ordem para o uso de cada elemento.

O Dublin Core qualificado inclui três elementos adicionais (Audiência, Proveniência e Detentor de Direitos), assim como um grupo de refinamentos de elementos (também chamados qualificadores), que refinam a semântica dos elementos de maneira que sejam úteis nas descobertas de recursos.

2.4 Workflow

A WfMC (1995) define *workflow* como sendo a facilitação ou automação computadorizada de um processo de negócio, em sua totalidade ou em parte. Ele está relacionado à automação de procedimentos durante o qual os documentos, informações ou tarefas são passadas de um participante para outro, de acordo com um conjunto de regras, para se alcançar ou contribuir com os objetivos gerais de negócio.

O *workflow* geralmente expressa as regras de negócio que podem existir. Cada negócio tem diferentes regras e políticas sobre as tarefas que devem ocorrer dentro da companhia (Mckay, 2004).

Um exemplo de regra de negócio seria: antes que uma página seja publicada, ela deve ser aprovada pela chefia, revisada por uma equipe técnica quanto corretude e questões legais e o texto deve ser corrigido quanto a erros gramaticais por um revisor especializado.

Workflow separa a lógica dessas regras de negócio e padroniza o conceito de se pensar nas mudanças. Com uma lógica separada, fica fácil para se mudar a aplicação para se adequar às regras de negócio. Frequentemente as aplicações tentam forçar a se usar um determinado *workflow*, pois o *workflow* foi codificado de maneira fixa na aplicação (Mckay, 2004).

2.5 Gestão de Conteúdo

Do ponto de vista de processo, Gestão de Conteúdo, em inglês *Content Management (CM)*, é a criação, publicação e gestão das informações e documentos da empresa na *web*. Isto pode ser feito por um *webmaster* na sede da empresa, ou distribuído pelos donos do conteúdo em cada filial (Mcnay, 2002).

Boiko (2005), tem uma definição mais ampla e afirma que Gestão de Conteúdo, significa coisas distintas para diferentes pessoas.

As diferentes perspectivas sobre o que a gestão de conteúdo faz para uma organização serve como uma definição do que é gestão de conteúdo.

- Na perspectiva de metas de negócio, CM distribui valores de negócio.
- Na perspectiva de análise, CM equilibra as forças organizacionais.
- Na perspectiva profissional, CM combina disciplinas relacionadas ao conteúdo.
- Na perspectiva de processo, CM coleta, gerencia e publica informação.
- Na perspectiva técnica, CM é infra-estrutura técnica.

Boiko diz que a gestão de conteúdo é e faz tudo isso.

2.6 Sistema de gestão de conteúdo

Um Sistema de Gestão de Conteúdo (*Content Management System* ou *CMS*) é uma ferramenta que permite a uma variada equipe técnica (centralizada) e equipe não técnica (descentralizada) a criar, editar, gerenciar e finalmente publicar (em diversos formatos) uma variedade de conteúdo (tais como, textos, gráficos, vídeos, etc), restringidas por um conjunto de regras, processos e *workflows* centralizados, que asseguram um conteúdo eletrônico coerente e validado (Contentmanager, 2008).

A idéia básica por trás de um CMS é a de separar o gerenciamento do conteúdo do design gráfico das páginas que apresentam o conteúdo. O design das páginas que apresentam os conteúdos é colocado em arquivos chamados moldes (*templates*), enquanto o conteúdo é armazenado em banco de dados ou arquivos separados. Quando um usuário solicita uma

página, as partes são combinadas para produzirem a página HTML padrão. A página resultante pode incluir conteúdos de diferentes fontes (Pereira e Bax, 2002).

O CMS deve permitir que os próprios colaboradores, no papel de autores, criem seus conteúdos sem necessidade de intermediários, utilizando os diversos programas disponíveis. Em seguida, estes conteúdos são armazenados em repositórios centralizados para serem tratados (gerenciados, padronizados, formatados e publicados no *website*) através do CMS. O CMS deve gerir também as revisões, atualizações e o controle de acesso, garantindo confiabilidade ao que será publicado e segurança quanto à propriedade e a autoria dos conteúdos (Pereira e Bax, 2002).

Um aspecto importante do CMS é a possibilidade de serem configurados processos simples de controle de fluxos de trabalho, para distribuição dos conteúdos nos canais de comunicação. Ao gerenciar a forma como os conteúdos são publicados, o CMS estabelece fluxos padronizados de trabalho, que definem os ciclos de vida desses conteúdos. Por exemplo, a empresa que anuncia seus produtos na *web* deverá retirar do sítio as promoções quando os estoques terminarem, sob pena de se obrigar a continuar vendendo aqueles produtos por preços indevidos (Pereira e Bax, 2002).

Um bom CMS deve tornar possível time ou colaborarem adicionar, editar e deletar conteúdos. Algumas das funcionalidades básicas cobertas são: esquemas de segurança baseados em papéis, sindicalização de conteúdo, suporte à criação de catálogos, indexação, busca e *workflow* (Pereira e Bax, 2002).

Em um nível mais amplo, um CMS é responsável por coletar, gerenciar e publicar fragmentos de informações conhecidos como componentes de conteúdo.

Boiko (2005), divide o CMS em três partes: o **sistema de coleta**, que transforma o conteúdo em componentes, o **sistema de gestão**, que é um tipo de banco de dados que armazena esses componentes e o **sistema de publicação** que captura os componentes do sistema de gestão e os transforma em publicações.

O **sistema de coleta** é responsável por todos os processos que ocorrem antes que o conteúdo esteja pronto para a publicação. Ele transforma informação crua em um conjunto bem organizado de componentes de conteúdo. Este processo inclui:

- **Autoria:** cria o conteúdo do zero.
- **Aquisição:** obtém o conteúdo de uma fonte existente.
- **Conversão:** remove as informações desnecessárias do conteúdo e altera as marcações da linguagem.
- **Agregação:** edita o conteúdo, divide em componentes e melhora-o para se adequar no sistema de metadados desejado.
- **Serviços de coleta:** programas de CMS e funções que ajudam no processo de coleta. Por exemplo, os serviços de coleta podem produzir formulários Web no qual se entra com o conteúdo para os componentes.

O **sistema de gestão** em um CMS é responsável pelo armazenamento a longo termo de um componente de conteúdo e uma gama de outros recursos. O sistema de gestão contém o repositório, o *workflow* e as ferramentas de administração. Em um nível mais alto, ele permite saber o que é coletado e qual é a sua disposição.

O sistema de gestão inclui:

- **Repositório:** um lugar para armazenar o conteúdo.
- **Administração:** um sistema de administração para instalar e configurar o CMS.
- **Workflow:** conjunto de passos definidos, para que seja feito o trabalho necessário sobre o conteúdo e torná-lo pronto para ser publicado.
- **Conexões:** um conjunto de conexões (hardware e software) para outros sistemas dentro da organização, abrangendo de redes e servidores até repositórios de dados.

O **sistema de publicação** é responsável por obter os componentes de conteúdo e outros recursos do repositório e criar publicações automaticamente a partir deles.

Um sistema de publicação inclui:

- **Templates de publicação:** programas que constroem publicações automaticamente.

- **Serviços de publicação:** um conjunto de ferramentas para controlar o que é publicado e como é publicado.
- **Conexões:** ferramentas e métodos usados para incluir dados de outros sistemas (não CMS) na publicação final.
- **Publicações Web:** a saída mais comum para a maioria dos CMS.
- **Outras publicações:** outras publicações não *web*, incluindo eletrônica, impressa e agregações (*syndications*).

2.7 Padrões Brasil e-gov

Os Padrões Brasil e-Gov são recomendações de boas práticas agrupadas em formato de cartilhas com o objetivo de aprimorar a comunicação e o fornecimento de informações e serviços prestados por meios eletrônicos pelos órgãos do Governo Federal. (Padrões Brasil e-GOV - Governo Eletrônico, 2008)

Seus objetivos são:

- Estabelecer padrões de qualidade de uso, desenho, arquitetura de informação e navegação;
- Estabelecer um fluxo de criação, desenvolvimento e manutenção na gestão dos sítios governamentais;
- Consolidar a acessibilidade;
- Criar artefatos de acordo com os padrões estabelecidos pelo W3C (World Wide Web Consortium).

As orientações do guia seguem quatro diretrizes:

1. **Páginas leves:** As páginas devem ter preferencialmente até 50 KB, somados código, conteúdo e imagens. Recomenda-se que as páginas não ultrapassem 70kb de tamanho;
2. **Separação da forma do conteúdo:** Separação da estrutura e o conteúdo da página

de sua codificação e visual. O posicionamento, a forma e a cor devem ser criados a partir de folhas de estilo (CSS);

3. **Páginas em conformidade com os padrões *web*:** Padrões *web* são especificações abertas que prevêm a acessibilidade desses documentos ao maior grupo de indivíduos possível;
4. **Páginas independentes de navegador e plataforma:** A utilização de elementos, atributos ou comandos proprietários podem comprometer, prejudicar e até impedir a visualização e o perfeito funcionamento da página em outros navegadores ou sistemas operacionais.

2.8 Parâmetros para analisar um CMS

Boiko (2005) cita um conjunto de passos para se comprar ou alugar um CMS.

- Criar uma visão geral de alto nível do projeto de CMS que se pretende implantar.
- Pesquisar no mercado por produtos que parecem se adequar ao projeto.
- Fazer um corte na lista de candidatos, selecionando aqueles que realmente valem à pena insistir.
- Enviar uma solicitação de proposta (*Request for Proposal*, RFP) para aqueles que passaram no primeiro corte.
- Selecionar um pequeno número de finalistas pontuando as respostas do RFP e qualquer resposta às perguntas feitas.
- Ter visitas técnicas rápidas e verificar as referências de cada um dos finalistas. Ter uma apresentação dos candidatos restantes.
- Tomar uma decisão final, comparando os pontos para as referências e apresentações com aqueles da RFP.

Segundo Cruz (2002), ao comprar um CMS, nenhuma empresa pode correr o risco de apenas e tão somente ater-se aos processos convencionais de soluções e estabelecer

matrizes comparativas de funcionalidades versus preços. O principal é, antes de tudo, “perguntar à empresa” que tipo de conteúdo e conhecimento ela precisa ou quer gerenciar.

Segundo os padrões e-gov (Padrões Brasil e-GOV - Governo Eletrônico, 2008) além da análise das necessidades, outros fatores a serem levados em conta são:

1. **O gerenciador cria páginas de acordo com os padrões *web*?** As perguntas “O gerenciador está de acordo com os padrões *web*?” e “Seus modelos nativos estão validados?” são fundamentais para verificar se o gerenciador de conteúdo está de acordo com os padrões *web*;
2. **O gerenciador cria páginas HTML acessíveis?** É importante lembrar que, apesar do gerenciador vir com estas características nativas, as mesmas podem ser perdidas na confecção dos modelos ou customizações no sistema;
3. **O gerenciador possui todos os elementos necessários à criação do sítio e/ou possui extensões que supram a necessidade do administrador?** É importante saber que recursos serão necessários no sítio: *chats*, ferramentas de fóruns e, se o gerenciador dispõem delas;
4. **O gerenciador de conteúdo possui uma boa rede de suporte? Fóruns e páginas especializadas onde se possam esclarecer as dúvidas?** Antes de adotar qualquer ferramenta é interessante fazer um estudo do material de apoio. As FAQs (*Frequently Asked Questions*) são fáceis de serem compreendidas? O fórum é ativo e as respostas às dúvidas são condizentes? Qual o apoio em português?
5. **O gerenciador permite a criação de múltiplos perfis?** O gerenciador permite a criação de perfis de publicação de acordo com as áreas, seções e níveis de responsabilidade (ex: usuário, editor, administrador)?
6. **Qual a complexidade do sistema?** Qual a curva de aprendizado do sistema? Qual a opinião de usuários sobre o uso do sistema?
7. **O gerenciador é capaz de criar URLs estáveis e amigáveis?** As URLs são estáveis, permanecendo a mesma com a alteração do conteúdo ou inserção de novos? O sistema possui um dispositivo para criação de URLs amigáveis? O

gerenciador responde de forma rápida? Qual a sua capacidade de requisições?

8. **Qual a carga de requisições que o sistema agüenta?** Qual a resposta do sistema em cada intervalo de número de requisições?

2.9 Considerações finais

Neste capítulo procurou-se dar uma visão mais abrangente sobre gestão de conteúdo, mostrando seus os conceitos básicos como a definição de dados, informação, conteúdo, metadados, *workflow* e as principais características de um sistema de gestão de conteúdo, bem como parâmetros para como analisá-los. Também se apresentou algumas diretrizes e recomendações específicas para o desenvolvimento de sítios de instituições federais, no qual o estudo de caso deste trabalho se enquadra.

Tais conceitos são fundamentais para o entendimento dos capítulos posteriores e serviram como base para a definição dos métodos apresentados no Capítulo 3.

Capítulo 3

Materiais e Métodos

3.1 Introdução

Neste capítulo é descrito os critérios adotados para a escolha do CMS. Nele é feita uma breve descrição das características do campus Francisco Beltrão da UTFPR, estudo de caso deste trabalho, e apresentado como foi feita a pesquisa dos CMS disponíveis no mercado e como se chegou aos três CMS finalistas. Também são listados quais os critérios adotados para a comparação destes CMS, que são aplicados nos capítulos 4 e 5.

3.2 Critérios

Para este trabalho, foram considerados apenas os CMS para a construção de sítios no formato de portal, não cobrindo aqueles que são específicos determinadas áreas, como *blogging*, *wikis*, comércio eletrônico, redes sociais, educação à distância. Além disso, pesquisou-se apenas CMS de código livre. Optou-se pelo uso de um CMS de código aberto pelo fato de serem gratuitos, livres para alteração e fornecerem uma base de usuários e suporte grande. A legislação para compras em instituições públicas, regida pela lei 8666/93, é feita através de processo licitatório, o que tornaria difícil a aquisição de um CMS proprietário.

Para a escolha do CMS procurou-se usar os passos sugeridos por Boiko, citados na seção 2.8. Entretanto, o conjunto de passos sugeridos por Boiko se refere à escolha de um CMS comercial e alguns deles, como enviar uma requisição proposta aos fornecedores ou visitar as empresas desenvolvedoras, não se aplica aos softwares livres. Entende-se que tais passos servem para conhecer melhor o produto a ser escolhido e neste trabalho foram substituídos por uma pesquisa mais aprofundada sobre o produto através da leitura da documentação e o teste do CMS.

Neste trabalho foram utilizados os seguintes passos:

- Criar uma visão geral de alto nível do projeto de CMS que se pretende implantar.
- Pesquisar no mercado por produtos que parecem se adequar ao projeto.

- Fazer um corte na lista de candidatos, selecionando aqueles que realmente valem à pena insistir.
- Fazer uma pesquisa mais aprofundada sobre os softwares, que passaram no primeiro corte e testá-los.
- Tomar uma decisão final, avaliando suas características e os recursos disponíveis.

3.3 Visão Geral do projeto de CMS

Para o desenvolvimento deste trabalho, tomou-se como estudo de casos a escolha de um CMS no campus Francisco Beltrão da Universidade Tecnológica Federal do Paraná (UTFPR).

O principal requisito deste caso é a construção de um sítio na internet no formato de um portal, onde são divulgadas as informações sobre o campus. As principais características são:

- As informações consistem em sua maior parte de texto, muitas vezes acompanhadas de imagens relacionadas ao assunto.
- O conteúdo da página pode ser em formato de notícias e eventos que têm relevância apenas durante um período de tempo ou páginas com conteúdo informacional sobre o campus que ficará permanente no sítio por um longo período, como o histórico, localização, estrutura organizacional, etc.
- As informações são geradas por diversas pessoas em diversos setores da instituição. O CMS deve possuir interface que permita que tais usuários não técnicos possam colaborar com o sítio, sem que eles precisem conhecer linguagens de marcação, como o HTML.
- Alguns setores podem ter necessidades especiais, portanto o CMS deve ser personalizável e extensível através de *plug-ins*.
- O CMS deve permitir níveis de permissões granulares para se poder delegar a atualização de determinadas áreas do sítio aos próprios membros encarregados por

aquele departamento.

- Os textos devem ser avaliados quanto a sua imparcialidade, relevância e correteude por uma ou mais equipes responsáveis pela edição. Portanto, existe a necessidade do software suportar *workflow*.
- Por se tratar de uma instituição federal, a página deve seguir as diretrizes de Padrões Brasil e-GOV para construção de sítios, incluindo padrões *web* e acessibilidade.
- O CMS deve ser de código aberto para permitir personalizações que eventualmente não possam ser feitas através de *plug-ins*.

3.4 Pesquisa sobre CMS disponíveis

Foi realizada uma pesquisa com alguns dos principais CMS de código aberto disponíveis. Nesta etapa foram excluídos os candidatos que não eram de código aberto e os que eram específicos para determinada áreas, como *blogs*, *wikis*, comércio eletrônico, redes sociais, educação à distância.

Foi obtida uma lista com 15 CMS de código aberto:

1. bitweaver
2. CMS Made Simple
3. Drupal
4. e107
5. eZ Publish
6. Joomla
7. Mambo
8. MODx
9. PHP-Fusion

10. PHP-Nuke

11. phpWebSite

12. Plone

13. Typo3

14. Website Baker

15. XOOPS

3.5 Corte na lista

Para diminuir o número de candidatos, procurou-se escolher os CMS que possuem melhor reputação no mercado. Para isso, classificou-se os CMS de acordo com a relevância da página em que os seus projetos estão hospedados, calculada a partir do algoritmo *PageRank*, utilizado pelo mecanismo de busca Google.

“A classificação das páginas (*PageRank*) confia na natureza excepcionalmente democrática da *Web*, usando sua vasta estrutura de *links* como um indicador do valor de uma página individual. Essencialmente, o Google interpreta um link da página A para a página B como um voto da página A para a página B. Mas o Google olha além do volume de votos, ou *links*, que uma página recebe; analisa também a página que dá o voto. Os votos dados por páginas ‘importantes’ pesam mais e ajudam a tornar outras páginas ‘importantes’” (Google, 2008b).

Como critério de desempate, utilizou-se a quantidade aproximada de páginas que contém um link para a página do CMS, que servem como um indicador da popularidade do CMS. Os dados foram obtidos através da busca avançada do motor de buscas do Google e estão representados no Quadro 3.1.

Posição	CMS	Página	Google PageRank	Quantidade de links
1°	Joomla	joomla.org	9	1.400.000
2°	Plone	Plone.org	9	166.000
3°	Mambo	mambo-foundation.org	9	13.200
4°	Drupal	drupal.org	8	90.700
5°	PHP-Fusion	www.php-fusion.co.uk	8	34.800

6°	XOOPS	Xoops.org	8	23.500
7°	eZ Publish	ez.no	8	12.600
8°	Typo3	typo3.org	8	5.070
9°	phpWebSite	phpwebsite.appstate.edu	7	7.340
10°	e107	E107.org	7	7.160
11°	MODx	modxcms.com	7	2.740
12°	Website Baker	websitebaker.org	6	4.640
13°	PHP-Nuke	phpnuke.org	5	452.000
14°	Bitweaver	bitweaver.org	5	11.600
15°	CMS Made Simple	cmsmadesimple.org	5	6.160

Quadro 3.1 - Classificação dos CMS de acordo com o PageRank e quantidade de links

Somente a quantidade de *links* para a página do CMS, não é um bom indicador da importância do CMS no mercado, pois este critério pode ser afetado por outros fatores. Um exemplo é o PHP-Nuke, cuja licença obriga que os seus utilizadores mantenham no rodapé de todas as páginas geradas as informações de direitos autorais e um link a página do PHP-Nuke. Para a remoção deste rodapé, o usuário deve comprar uma licença comercial (Php-Nuke, 2008). Este termo na licença explica a grande quantidade de *links* referenciando para a sua página, apesar deste ter um PageRank relativamente baixo.

Escolheu-se os três primeiros classificados de acordo com o Quadro 3.1 para se realizar uma pesquisa mais aprofundada. Durante esta pesquisa, descobriu-se que o Joomla é um produto criado a partir do código fonte do Mambo. Devido a divergências ideológicas, alguns dos desenvolvedores do Mambo resolveram se separar em 2005 e criar o Joomla. Ambos os produtos possuem características muito semelhantes e para evitar a comparação entre produtos muito parecidos, decidiu-se excluir o Mambo da avaliação, dando lugar ao quarto classificado, o Drupal. Deu-se preferência ao Joomla como candidato finalista, pois além da classificação melhor que o Mambo no Quadro 3.1, verificou-se que desde a data de criação do Joomla, em 2005, o interesse do público pelo Mambo vem diminuindo. Esta tendência pode ser observada no gráfico da Figura 3.1, que mostra o volume de buscas no Google pelos termos “mambo” e “joomla” na categoria Internet, desde 2004 até o presente. (Google, 2008a).

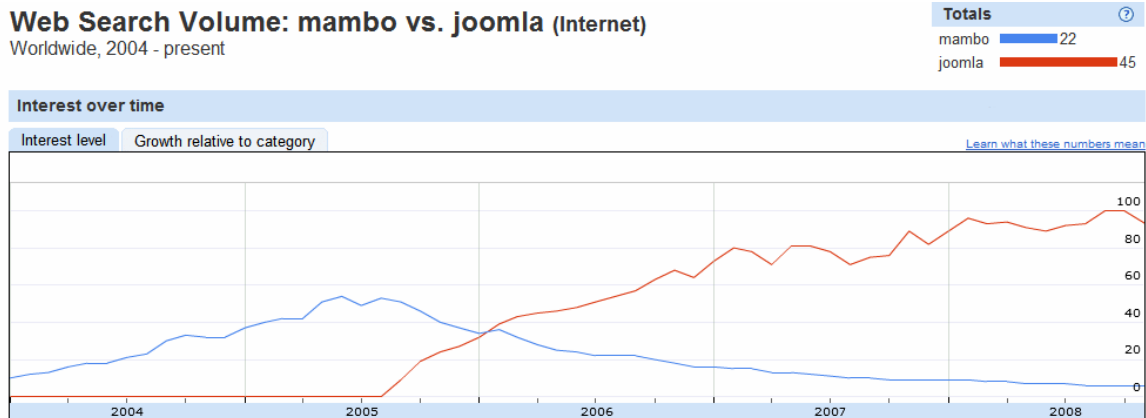


Figura 3.1 - Volume de buscas com os termos “mambo” e “joomla” (Google, 2008a).

3.6 Comparação entre os CMS

Foi feita uma pesquisa mais detalhada e uma comparação entre os recursos dos CMS selecionados Joomla, Plone e Drupal. Nesta comparação, foram analisados critérios técnicos, utilização de padrões *web*, acessibilidade, *workflow*, internacionalização e localização, facilidade de uso, plug-ins, segurança e documentação. Os resultados são apresentados nos Capítulos 4 e 5.

3.6.1 Critérios Técnicos

- **Sistema operacional:** Quais os sistemas operacionais em que ele roda?
- **Servidor de aplicações:** Qual o servidor de aplicação em que ele roda?
- **Servidor Web:** Em quais servidores *Web* ele roda?
- **Banco de Dados:** Quais os bancos de dados que ele suporta?
- **Linguagem:** Em que linguagem de programação o CMS foi escrito?
- **Licença:** Qual o tipo de licença sobre o qual ele foi lançado?
- **Acesso Root:** É necessário privilégios de administrador para ele funcionar ou basta um usuário comum?

- **Acesso Shell:** É necessário acesso ao *shell* do sistema operacional ou apenas acesso aos arquivos via FTP (*File Transfer Protocol*) seria suficiente?

3.6.2 Utilização de padrões Web, e-gov e acessibilidade

Foi verificado se as páginas utilizam de padrões *Web* como XHTML, CSS. Para isso, utilizou-se as ferramentas de validação da W3C.

Verificou-se também se o sítio segue as diretrizes dos padrões Brasil e-gov como URLs amigáveis e se possuem recursos de acessibilidade.

Apesar de alguns CMS afirmarem que estão de acordo com as Diretrizes de acessibilidade ao conteúdo da *Web* (*Web Content Accessibility Guidelines* - WCAG) da W3C, estas afirmações não puderam ser comprovadas. O WCAG 1.0 engloba 14 diretrizes contendo princípios gerais para projetos acessíveis. Nem todas as diretrizes podem ser verificadas por ferramentas automatizadas. Uma análise de acessibilidade feita por humanos exigiria um conjunto de testes mais aprofundados, envolvendo o uso de diferentes navegadores, diferentes configurações, utilização de sintetizador de voz e o teste do sítio por pessoas com deficiências.

3.6.3 Workflow

Avaliou-se se o software de CMS suporta níveis de privilégios de usuários baseado em grupos e *workflow*.

3.6.4 Internacionalização e Localização

Como a maioria dos CMS é feito para utilizadores de língua inglesa, foi verificado se eles possuem traduções para o português e também suporte a localização, como formato de datas, números, moedas, etc.

3.6.5 Facilidade de Uso

Verificou-se se o CMS é fácil de usar por usuários não técnicos. O principal item observado foi se ele possui editor visual WYSIWYG (*What You See is What You Get*), ou seja, um editor que permita que se formate o texto, insira *links* e imagens sem que seja

necessário que o usuário saiba HTML ou outra linguagem de marcação. Também foram analisados outros recursos que facilitem o seu uso como o *drag n'drop* (arrastar e soltar itens), redimensionador de imagens, botão desfazer, etc.

3.6.6 Plug-ins

Verificou-se caso o CMS não possua algum recurso nativamente, se ele possui plug-ins que adicionam tal funcionalidade. Procurou-se, entretanto, escolher um CMS que possua a maioria dos recursos nativamente, uma vez que plug-ins desenvolvidos por terceiros, nem sempre possui a mesma qualidade do CMS em si, muitos são desenvolvidos por voluntários que colaboram com o projeto em seu tempo livre e não se tem garantia de suporte e de que eles continuarão sendo atualizados.

3.6.7 Segurança

Analisou-se a quantidade e a gravidade das vulnerabilidades que foram descobertas nos CMS e reportadas no sítio especializado em segurança Secunia (Secunia, 2008).

Critérios de segurança como criptografia de senha durante o *login* não foi considerado, pois se considera tarefa do servidor *web*, através do uso de SSL, e não do CMS.

3.6.8 Documentação e suporte

Analisou-se a documentação do CMS, se ela cobre todos os pontos do CMS e se existem traduções dela para o português. Além disso, verificou se eles possuem listas de *email* e fóruns para discussões. Pesquisou-se a quantidade de livros dedicados aos CMS que foram publicados e se existem livros em português sobre o assunto.

Capítulo 4

Análise dos CMS

4.1 Introdução

Neste capítulo é feita a análise individual de cada um dos CMS analisados: Drupal, Joomla e Plone. Nele são mostrados alguns pontos como: características técnicas, padrões *web* e acessibilidade, *workflow*, internacionalização e localização, facilidade de uso, *plug-ins*, segurança, documentação e suporte.

4.2 Drupal

4.2.1 Características técnicas

O Drupal é um CMS que tem como destaque a simplicidade. Ele é formado por um núcleo com apenas funcionalidades básicas e todos os outros recursos são providos através de módulos. Mesmo os seus recursos básicos de gestão de conteúdo são fornecidos como módulos chamados de essenciais, que não podem ser desativados. Em sua instalação ele acompanha outros módulos opcionais que estendem a sua funcionalidade.

O Drupal é CMS multiplataforma e roda em ambiente Unix, Linux ou Windows. A sua atual versão 6 roda sobre o PHP 4.3.5 e banco de dados MySQL 4.1. A sua versão 7, que está em desenvolvimento, exigirá PHP 5.2 e MySQL 5.0. O Banco de dados PostgreSQL também é suportado, entretanto, alguns módulos de terceiros podem não funcionar corretamente no PostgreSQL. Os seus desenvolvedores recomendam o servidor Apache, embora o Microsoft IIS, também funcione.

Sistema operacional	Unix, Linux, Windows
Servidor de aplicações	PHP 4.3.5 ou superior
Servidor <i>Web</i>	Apache, IIS
Banco de dados	MySQL 4.1 ou superior, PostgreSQL 7.4 ou superior
Linguagem	PHP
Licença	GNU GPL
Acesso <i>Root</i>	Não é necessário.
Acesso Shell	Não é necessário.

Quadro 4.1 - Detalhes técnicos do Drupal

4.2.2 Padrões Web e Acessibilidade

As páginas geradas pelo Drupal têm como tipo de documento XHTML 1.0 e utiliza folhas de estilo CSS para separar o conteúdo da apresentação. As páginas geradas pela instalação com o tema padrão do Drupal são validadas corretamente pela ferramenta de validação do W3C. Entretanto, a validação do código pode parar de funcionar, conforme o tema do sítio e os módulos que ele roda. O próprio sítio do Drupal utiliza um tema que não é validado corretamente. No código HTML de sua página existe um comentário alertando que ele não valida, que eles gostariam que validasse, mas isso significaria uma experiência do usuário reduzida para a maioria de seus visitantes, conforme o fragmento do código da Figura 4.1.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <!-- Note: does not validate. We would like it to, but that would mean
reduced user experience for the majority of our visitors. -->
```

Figura 4.1 - Fragmento de código do sítio drupal.org

O Drupal possui o recurso de URLs amigáveis, desde que o Apache tenha instalado o módulo `mod_rewrite`. Para a utilização deste recurso no Microsoft IIS, é necessário um instalar uma extensão de terceiros. O uso das configurações do `mod_rewrite` deixa as URLs legíveis, mas não permite a personalização delas, para evitar essa limitação, é necessário instalar um módulo do Drupal chamado “*Path*”.

O Drupal não possui recursos extras de acessibilidade, tais como ampliar fonte, contraste ou teclas de atalho pelo teclado. Seus únicos recursos de acessibilidade são os temas escritos por contribuintes que focam ser mais acessíveis para usuários deficientes visuais. Ao contrário do outros dois CMS analisados, os desenvolvedores não declaram publicamente o comprometimento em se tentar atingir o padrão WCAG.

4.2.3 Workflow

O controle de usuários no Drupal é feito através de grupos, também chamado de papéis, que agrupam um conjunto de permissões que podem ser aplicado aos usuários do sítio.

Em sua instalação padrão, o Drupal não apresenta suporte para *workflow*. Ele permite

adicionar suporte para *workflow* através de um módulo separado chamado “*workflow*” que pode ser baixado em seu sítio.

4.2.4 Internacionalização e Localização

O Drupal suporta mais de 40 idiomas, no entanto, alguns deles não estão atualizados para a versão mais atual do Drupal. As traduções não vêm na instalação padrão e devem ser baixada separadamente. Durante a instalação do CMS, o *wizard* de instalação fornece as instruções de como baixar e instalar a tradução, que consiste em descompactar o pacote de tradução no diretório da instalação do sítio.

4.2.5 Facilidade de uso

O Drupal não possui um editor visual WYSIWYG em sua instalação padrão. É necessário que o usuário saiba HTML para formatar o conteúdo. Porém, existem diferentes módulos que oferecem tal funcionalidade ao Drupal, dentre eles se destacam o TinyMCE e o FCKeditor.

O TinyMCE possui mais recursos que o FCKeditor, incluindo corretor ortográfico, permissões de quem pode usar o editor e recursos de acessibilidade como teclas de atalhos para acessar o menu. Para o *upload* de imagens, é necessário instalar outro *plug-in*, chamado Image Assist.

O FCKeditor é mais simples e não exige configurações extras. Um recurso interessante do FCKeditor é a opção “colar do Word”, que permite colar textos escritos no Microsoft Word, removendo o HTML que contém as configurações de estilos e tamanho de fontes, mas mantendo a formatação, como negrito, itálico.

Ambos os editores possuem recursos de desfazer ações e de redimensionar imagens. Entretanto, nos testes realizados, nenhum deles conseguiu gerar XHTML válido para textos com formatações mais complexas.

4.2.6 Plugins

O Drupal fornece um núcleo com os recursos básicos de um CMS e possui *plug-ins* ou módulos que estendem a sua funcionalidade. Sua versão 6.2 possui com 33 módulos

núcleo, sendo 5 essenciais, que não podem ser desativados, e 28 módulos opcionais. Além disso, ele possui centenas de módulos gratuitos criados por colaboradores, disponível para *download*. Para a instalação destes módulos, basta descompactá-los no diretório *modules* do Drupal e ativá-lo no painel de administração do sítio, alguns deles exigem a execução de scripts de instalação para a configuração do banco de dados.

4.2.7 Segurança

De acordo com a Secunia (Secunia, 2008), desde 2003, foram emitidos 10 alertas de segurança que endereçam 22 falhas de segurança no Drupal. Até Novembro de 2008, todas elas estavam corrigidas. 60% dos problemas foram considerados menores e 40% moderados (Figura 4.2).

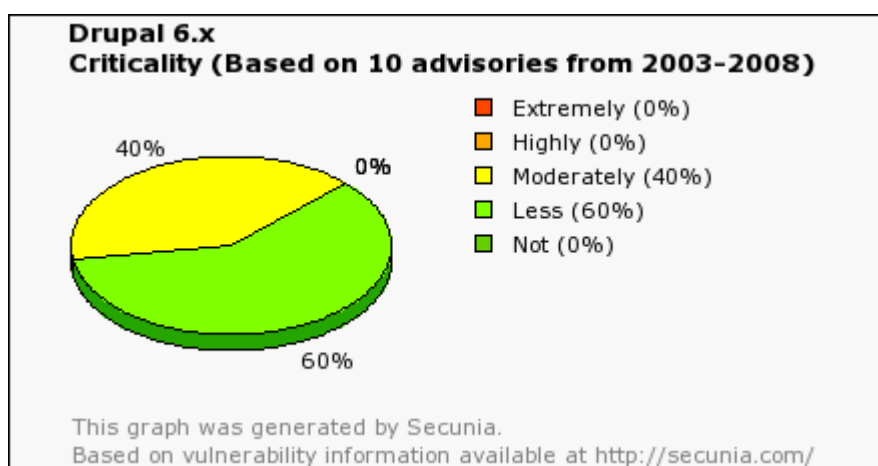


Figura 4.2 - Gravidade dos problemas de segurança do Drupal (Secunia, 2008).

90% das falhas do Drupal podiam ser exploradas remotamente e 10% através da máquina local (Figura 4.3).

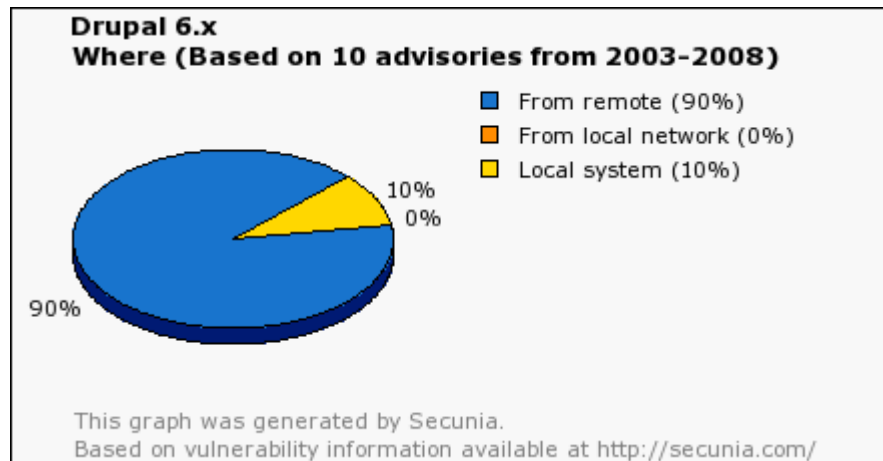


Figura 4.3 - Onde as falhas do Drupal poderiam ser exploradas (Secunia, 2008).

A maior parte das falhas do Drupal eram em relação à manipulação de dados (28%), *Cross Site Scripting* (28%) e contorno de segurança (22%) (Figura 4.4).

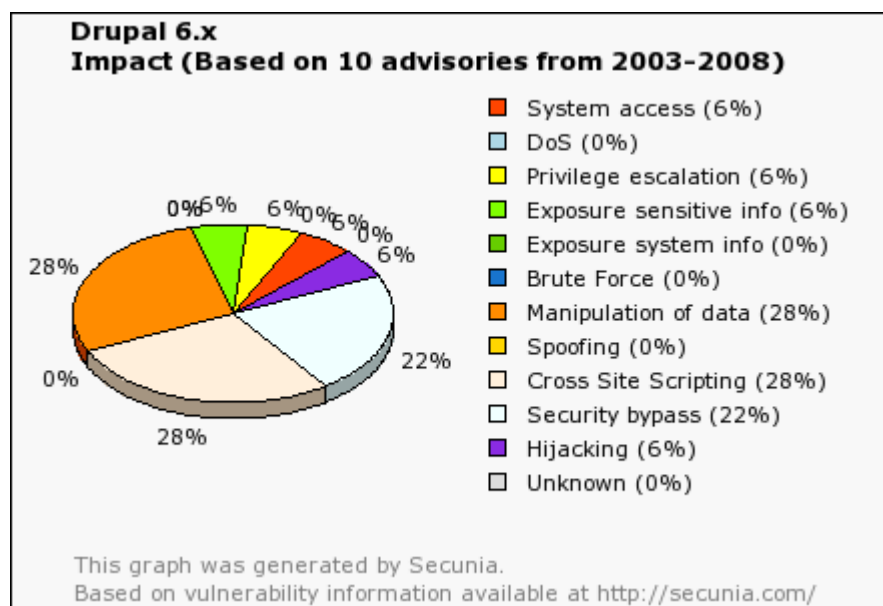


Figura 4.4 - Impacto das falhas de segurança do Drupal (Secunia, 2008).

4.2.8 Documentação e suporte

O Drupal possui sua documentação *online* e suporte aos usuários através de fóruns, listas de discussões e bate-papo através do protocolo IRC. Todo o suporte e documentação são fornecidos por colaboradores e voluntários do projeto. O suporte e a documentação em português estão disponíveis pelo sítio Drupal Brasil, que também mantém a tradução do

projeto.

Foram encontrados 20 títulos de livros publicados ou a serem publicados em 2009 que têm como assunto o Drupal, até o momento, nenhum deles possui tradução para o português. A lista completa dos livros se encontra no Apêndice B.

4.3 Joomla

4.3.1 Características técnicas

O Joomla roda em ambiente Unix, Linux ou Windows. A versão analisada foi a 1.5 cujo requisito mínimo é PHP 4.2, banco de dados MySQL 3.23 e servidor Apache 1.3. Embora outros servidores *Web* e banco de dados possam funcionar com algumas adaptações nas configurações, eles não são suportados oficialmente pelo Joomla.

Sistema operacional	Unix, Linux, Windows
Servidor de aplicações	PHP 4.2 ou superior
Servidor Web	Apache 1.3 ou superior
Banco de dados	MySQL 3.23 ou superior
Linguagem	PHP
Licença	GNU GPL v2
Acesso <i>Root</i>	Não é necessário.
Acesso Shell	Não é necessário.

Quadro 4.2 - Detalhes técnicos do Joomla

4.3.2 Padrões Web e Acessibilidade

O Joomla gera páginas em XHTML e utiliza de folhas de estilos CSS. Ambos foram validados corretamente pelas ferramentas de validação da W3C. Em seu manual, existe uma declaração em que se afirma o comprometimento de se seguir os padrões web e de se atingir o nível de acessibilidade WCAG prioridade 2 (Joomla, 2008). Dois dos três temas que vêm na instalação padrão do Joomla possui o recurso de tamanho da fonte para auxiliar usuários com baixa visão.

Assim como o Drupal, o Joomla suporta uso de URLs amigáveis através do `mod_rewrite` do Apache.

4.3.3 Workflow

Da mesma maneira que o Drupal, o Joomla não tem suporte nativo para *workflow*. Existe apenas um controle básico de permissões que é feito através de grupos, contendo sete níveis de acessos, sendo quatro para o *frontend* (registrado, autor, editor, redator) e três para o *backend* (gerente, administrador, super administrador). Não é possível criar novos grupos ou editar os existentes para se adequar as necessidades para um caso em particular. O suporte para ACL (*Access Control List*), está prevista para a versão 1.6 do Joomla.

Para adicionar um melhor controle de acessos ao Joomla, é necessária a compra de um componente comercial como o JUGA, JAACL Plus ou o cACL. Tais extensões fazem apenas o controle de acesso, permitindo criar novos grupos e permissões, mas não dá suporte para *workflow*.

Existe uma extensão chamada oWorkflows que adiciona a funcionalidade de *workflow* ao Joomla. Para não ficar limitado aos 7 grupos padrões do Joomla, ele permite a integração com os grupos do JUGA e do JAACL. Entretanto, a versão 0.3.7s do oWorkflows oferecida atualmente ainda não suporta a versão do Joomla 1.5.

Até novembro de 2008, momento da escrita deste trabalho, não há uma solução gratuita para um controle avançado de usuários no Joomla e uma solução para *workflow* que funcione com a versão do Joomla analisada.

4.3.4 Internacionalização e Localização

Antes da versão 1.5 do Joomla, apenas a interface do usuário suportava outros idiomas além do inglês. A partir da versão 1.5, as traduções das interfaces do Joomla estão divididas em três partes: a interface de instalação, a interface de administração e o sítio em si. A interface de instalação do Joomla, que guia o administrador a montar e configurar o sítio, está disponível em 51 idiomas. Após a instalação, para traduzir o sítio e a interface de administração é necessário baixar os pacotes de tradução e instalá-los. No sítio de extensões do Joomla estão disponíveis 98 pacotes de idiomas para *download*.

4.3.5 Facilidade de uso

O Joomla possui como padrão o editor visual TinyMCE 2.0, que como no caso do Drupal,

não gera códigos válidos. A instalação do Joomla possui como alternativa o editor XStandard Lite que foca em oferecer páginas seguindo padrões *web* gerando XHTML válidos e utilizando as classes definidas nos CSS para a formatação dos textos. Para a sua utilização é necessário fazer um cadastro no sítio do XStandard e instalar um *plug-in* no navegador. Além disso, para a sua instalação, é necessário que o usuário tenha permissões de administrador em sua estação de trabalho e rode o sistema operacional Windows ou Mac OS/X. Esses requisitos tiram de certa forma o conceito de se poder gerenciar o conteúdo de qualquer lugar, pois máquinas rodando Linux ou que não possuem privilégios para instalação de programas não poderão utilizar este WYSIWYG.

Apesar de gerar um código limpo e válido, seus recursos de formatação são inferiores do que o TinyMCE. Seu uso é mais adequado para se usar em conjunto com um editor de textos externo como o Word, onde o texto é formatado usando este editor e o XStandard filtraria as *tags* HTML do texto colado, convertendo-o para a formatação válida.

4.3.6 Plugins

As extensões do Joomla são classificadas em três tipos: Componentes, Módulos e Plugins. Os componentes rodam a parte principal de uma página, como artigos, *blogs*, sessões, informações de perfil dos usuários, etc. Os módulos são pedaços de informações secundárias que fazem parte do sítio, como menus, caixas de *login*, *banners*, últimas notícias, etc. Os *plugins* são pedaços de código que executam alguma lógica e geralmente não interagem com o usuário.

O Joomla possui um diretório com centenas de extensões para *download* e, diferentemente dos outros dois CMS analisados, além das extensões gratuitas, o Joomla possui um grande número de extensões comerciais.

4.3.7 Segurança

Segundo o Secunia, desde 2003, foram emitidos 19 alertas de segurança que endereçam 66 falhas de segurança no Joomla. Até Novembro de 2008 todas elas estavam corrigidas ou por um remendo de correção (89%) ou por uma forma contorno (11%), conforme a Figura 4.5.

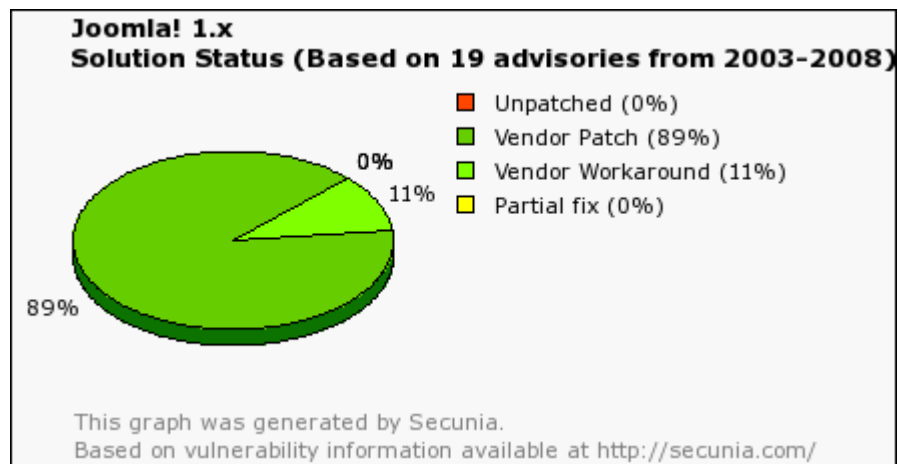


Figura 4.5 - Soluções para os problemas de segurança do Joomla

63% dos problemas foram considerados moderados, 32% menores e 5% não críticos (Figura 4.6).

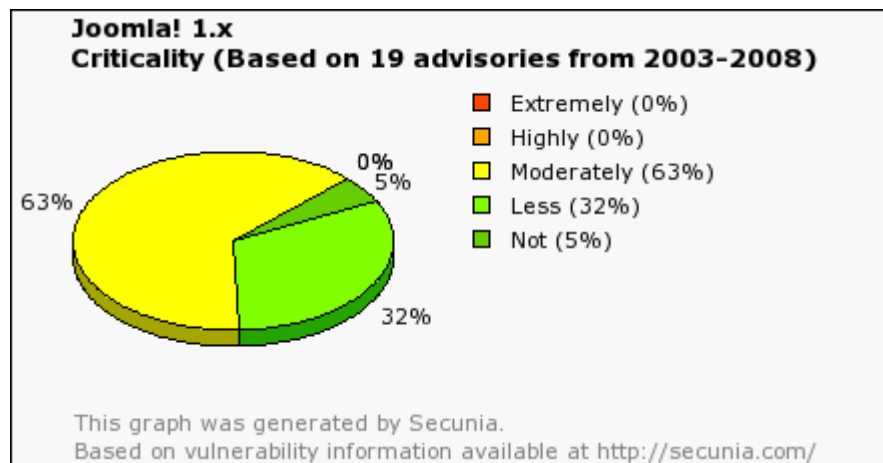


Figura 4.6 - Gravidade dos problemas de segurança do Joomla.

Todas as falhas de segurança do Joomla eram exploráveis remotamente (Figura 4.7).

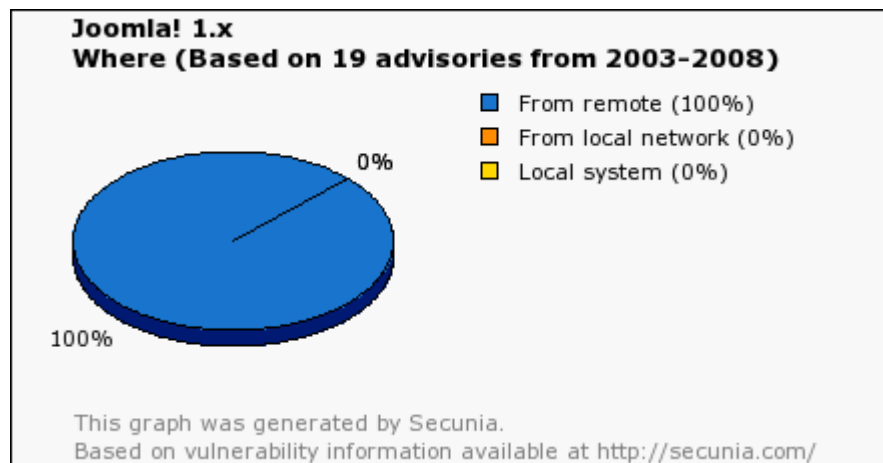


Figura 4.7 - Onde as falhas do Joomla poderiam ser exploradas

A maior parte das falhas de segurança foram relacionados a *Cross Site Scripting* (30%), manipulação de dados (24%), desconhecidos (15%) e contorno de segurança (12%), conforme a Figura 4.8.

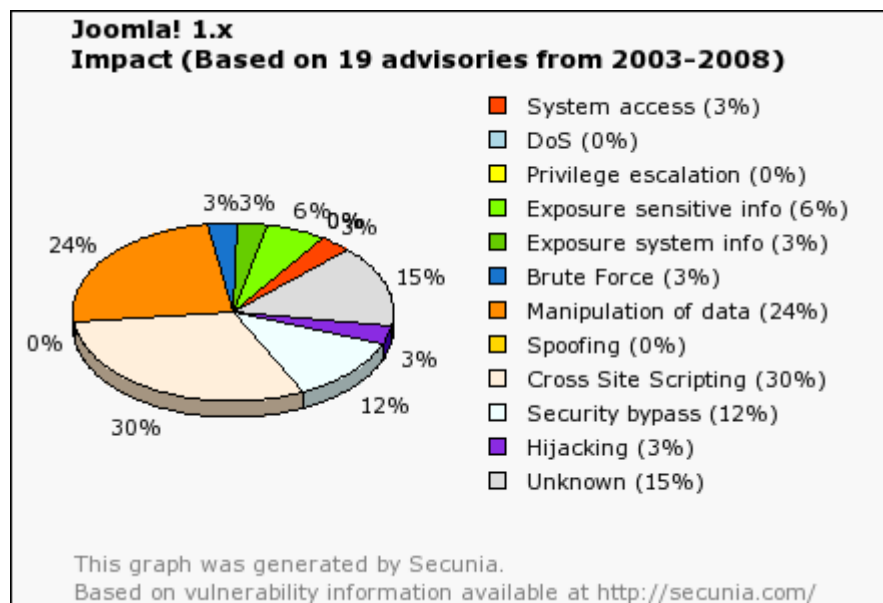


Figura 4.8 - Impacto das falhas de segurança do Joomla

4.3.8 Documentação e suporte

O Joomla possui a sua documentação on-line e o único suporte oficial aos usuários é feito através do fórum em seu sítio. Embora seus usuários tenham criado listas de e-mails para discussão, não existe uma lista oficial suportada por seus desenvolvedores. Existe também um canal de bate-papos pelo IRC com muitos de seus operadores pertencentes à equipe de desenvolvimento do Joomla, mas este também não é considerado oficial para termos de

suporte.

Recursos de documentação e suporte em língua portuguesa podem ser encontrados nos sítios Joomla Brasil e JoomlaBR, além de um subfórum dedicado a língua portuguesa no sítio oficial do Joomla.

Foram encontrados 26 títulos de livros publicados ou a serem publicados em 2009 que têm como assunto o Joomla, apenas o livro “Joomla – Guia do Operador” se encontra em língua portuguesa. A lista completa dos livros se encontra no Apêndice B

4.4 Plone

4.4.1 Características técnicas

O Plone é um sistema de gestão de conteúdos construído em cima do servidor de aplicações Zope, que é escrito em linguagem Python e faz a persistência de seus objetos de forma transparente em um banco de dados orientado a objetos próprio, chamado ZODB (*Zope Object Data Base*). O Plone roda sobre as plataformas Windows, Unix e Linux. Alguns de seus recursos administrativos exigem que se tenha acesso *Shell* na máquina onde o Zope esteja instalado, entretanto a maioria das tarefas de gerência de conteúdos pode ser feita através no navegador *Web*.

Sistema operacional	Unix, Linux, Windows
Servidor de aplicações	Zope
Servidor <i>Web</i>	Apache, IIS (ou qualquer um que funcione como <i>proxy</i>)
Banco de dados	ZODB
Linguagem	Python
Licença	GNU GPL
Acesso <i>Root</i>	Não é necessário.
Acesso <i>Shell</i>	É necessário.

Quadro 4.3 - Detalhes técnicos do Plone

4.4.2 Padrões Web e Acessibilidade

O Plone foi o primeiro CMS 100% compatível com os padrões XHTML e CSS, funcionando em qualquer dispositivo que suporte HTML, incluindo navegadores antigos e aparelhos celulares. Também foi o primeiro CMS compatível com os padrões de acessibilidade WAI-AA e US Section 508, permitindo que usuários com deficiências

visuais ou motoras possam usar o sítio (Plone, 2008).

O código das páginas do Plone facilita a sua leitura por softwares leitores de tela usados por deficientes visuais e possui atalhos de teclado que ajudam pessoas com dificuldade de usar o mouse, ou mesmo quem acessa o sítio através de um aparelho celular.

O Plone permite URLs amigáveis completamente personalizáveis pelo navegador, não havendo a necessidade de editar arquivos de configuração do mod_rewrite do Apache, como nos outros CMS analisados.

O Plone foi o único dos CMS analisados que suporta o padrão Dublin Core para metadados, sem a necessidade de instalar nenhum *plug-in*.

4.4.3 Workflow

O controle de acesso no Plone é feito através de permissões e níveis de acesso de forma granular. As permissões podem ser globais para o sítio, ou específicas para alguma seção do sítio. Ele permite a criação de grupos que podem agrupar um conjunto de regras de permissões, para serem aplicadas mais facilmente a vários usuários.

O Plone possui suporte nativo a *workflow* e permite regras de *workflow*, não só pelo tipo de conteúdo, como também pela localização do conteúdo no sítio. Desta maneira, diferentes partes do sítio podem ter diferentes configurações de segurança e diferentes caminhos de *workflow* para os mesmos tipos de objetos.

4.4.4 Internacionalização e Localização

O Plone está disponível em mais de 35 idiomas e foi projetado desde o início para gerenciar conteúdos de páginas multilíngües. Diferentemente dos outros dois CMS avaliados, a instalação do Plone já acompanha as traduções para todos os idiomas suportados, não sendo necessário baixá-las separadamente.

4.4.5 Facilidade de uso

O Plone possui um editor visual padrão chamado Kupu, que gera códigos de páginas XHTML válidas. Diferentemente dos outros WYSIWYG, ele não permite que o usuário

use código de formatação, como cores e tamanhos de fontes, misturado com o conteúdo. Ao invés disso, ele só permite o uso de estilos pré-definidos no CSS. Apesar de isso limitar as possibilidades de formatações diferentes que podem ser usadas no sítio, ela garante a consistência da formatação entre as páginas, pois apenas os estilos e cores que o *webmaster* escolher poderão ser usados no sítio.

O Kupu se integra com o Plone, permitindo inserção e redimensionamento de imagens. Os *links* internos das páginas podem ser inseridos a partir uma lista, evitando que o usuário digite o endereço errado e se tenha *links* quebrados. O Plone mantém a integridade dos *links* para as páginas internas, de modo que se uma página tem o seu endereço alterado, todas as outras que possuem alguma referência para ela, são atualizadas.

Apesar de o Plone ser fácil de usar pelo usuário final, a configuração e personalização do Plone pelo administrador do sítio é mais complicado que os outros dois CMS. É necessário conhecer como funciona a estrutura de objetos, camadas e *templates* do Zope, exigindo um tempo de aprendizado maior que os outros CMS. Além disso, o uso da linguagem Python, menos popular que o PHP e não suportada pela maioria dos serviços de hospedagem de sítios mais baratos, faz com que apenas empresas maiores, que possuem servidor próprio, ou que possam pagar a mais para ter um servidor que suporte o Zope, o adotem.

4.4.6 Plugins

O diretório de extensões do Plone conta com 926 projetos com 2406 lançamentos, divididos em 28 categorias. A maneira tradicional de se instalar um *plug-in* nas versões anteriores do Plone era baixar o pacote no sítio e descompactar o arquivo na pasta “Products” do Plone, maneira semelhante à instalação de *plug-ins* no Drupal. A partir do Plone 3, adotou-se uma nova forma de se instalar os *plug-ins*, utilizando-se pacotes Python *eggs*, que é a maneira padrão como os pacotes Python são distribuídos. No Plone, os *eggs* podem ser instalados a partir de um script chamado “*buildout*”, que pode automatizar a instalação de vários pacotes e configurações. Esta maneira é útil para automatizar a implantação de um sistema completo e configurado em um ambiente de produção. Outra maneira de se instalar os *eggs* é através do utilitário *easy_install*, que instala pacotes individualmente de forma rápida. Uma das vantagens da utilização dos *eggs* é que o utilitário baixa automaticamente os pacotes da internet e as suas eventuais dependências,

por outro lado, existe a desvantagem de ser necessário acesso *shell* a máquina onde a instância do Plone está instalada. Muitos dos desenvolvedores de *plug-ins* ainda não adotaram a distribuição de seus pacotes na forma de *eggs*, fazendo com que o diretório de extensões do Plone tenha duas formas distintas de se instalar *plug-ins*, que pode complicar a tarefa de administração de um sítio com vários *add-ons* instalados.

4.4.7 Segurança

Segundo o Secunia, desde 2003, foram emitidos 3 alertas de segurança que endereçam 4 falhas de segurança no Plone 3.x. Até Novembro de 2008 todas elas estavam corrigidas. Estas falhas foram classificadas como altamente crítica (33%), moderadamente crítica (33%) e menos crítica (33%), conforme a Figura 4.9

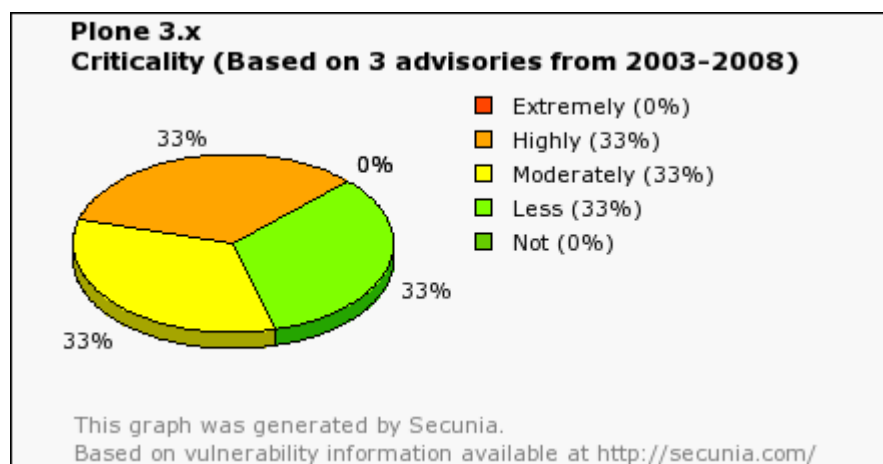


Figura 4.9 - Gravidade dos problemas de segurança do Plone

Todas as falhas descobertas podiam ser exploradas remotamente, conforme a Figura 4.10.

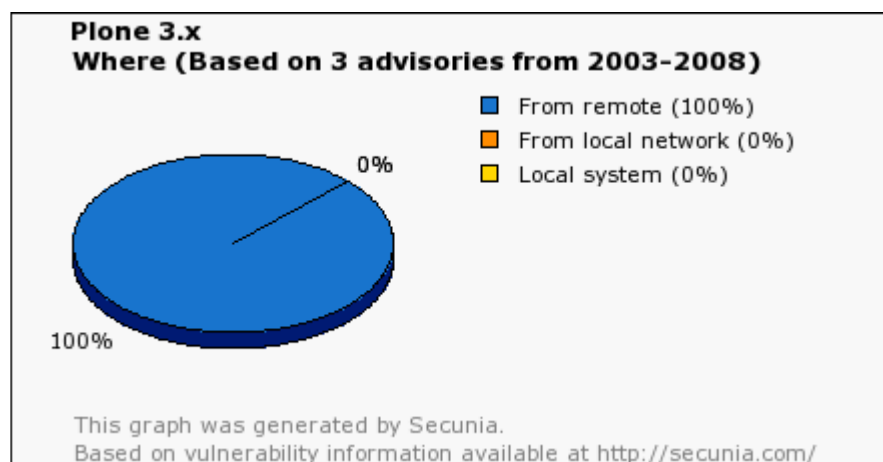


Figura 4.10 - Onde as falhas do Plone poderiam ser exploradas

Dentre as falhas descobertas, 67% se referiam a *cross site scripting* e 33% de acesso ao sistema (Figura 4.11).

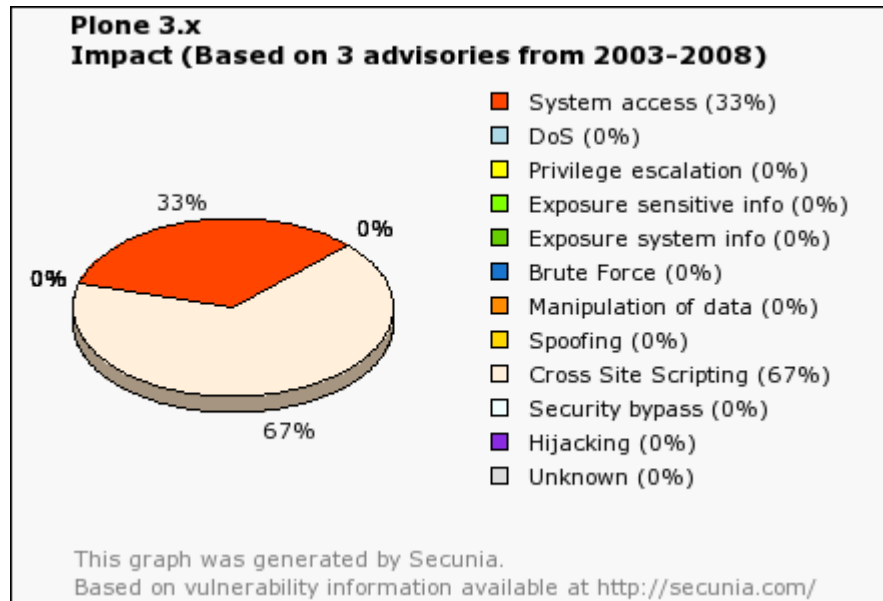


Figura 4.11 - Impacto das falhas de segurança do Plone

4.4.8 Documentação e suporte

O Plone possui sua documentação *online* e suporte aos usuários através de fóruns, listas de discussões e bate-papo através de um *applet* Java em seu sítio ou pelo protocolo IRC. O principal recurso em língua portuguesa para o Plone é o sítio Tchê Zope, onde se pode encontrar traduções de documentações, artigos, tutorias e fórum de discussão. Foram encontrados 12 títulos de livros publicados ou a serem publicados em 2009 que têm como assunto o Plone, até o momento, nenhum deles possui tradução para o português. A lista completa dos livros se encontra no Apêndice B.

Capítulo 5

Comparação dos CMS

5.1 Introdução

Este capítulo faz um paralelo das características dos CMS apresentadas no capítulo anterior, com o auxílio de quadros comparativos.

5.2 Características técnicas

Todos os CMS avaliados são multiplataforma. O Drupal e o Joomla possuem características técnicas semelhantes, exigindo configurações de servidores mais simples, utilizando Apache, PHP e MySQL. O Plone perde neste quesito por ser escrito em linguagem Python, que é menos comum no desenvolvimento *web*, exigir que se tenha o servidor Zope paralelo ao Apache e requerer que se tenha acesso *shell* ao servidor para algumas ações de administração (Quadro 5.1).

	Drupal	Joomla	Plone
Sistema operacional	Unix, Linux, Windows	Unix, Linux, Windows	Unix, Linux, Windows
Servidor de aplicações	PHP 4.3.5 ou superior	PHP 4.2 ou superior	Zope
Servidor Web	Apache, IIS	Apache 1.3 ou superior	Apache, IIS (ou qualquer um que funcione como <i>proxy</i>)
Banco de dados	MySQL 4.1 ou superior, PostgreSQL 7.4 ou superior	MySQL 3.23 ou superior	ZODB
Linguagem	PHP	PHP	Python
Licença	GNU GPL	GNU GPL v2	GNU GPL
Acesso Root	Não é necessário.	Não é necessário.	Não é necessário.
Acesso Shell	Não é necessário.	Não é necessário.	É necessário.

Quadro 5.1 – Comparação dos critérios técnicos

5.3 Padrões Web e acessibilidade

Os *templates* originais dos três CMS utilizam-se dos padrões XHTML e CSS, entretanto, apenas o Plone consegue manter a validade do XHTML em páginas geradas a partir do seu

editor visual, sem que seja instalado nada no lado do cliente.

Todos os CMS analisados são capazes de gerar URL amigáveis, mas apenas o Plone possui esta opção nativa e habilitada por padrão. Somente o Plone utiliza o padrão Dublin Core para metadados.

Quanto aos recursos de acessibilidade, o Joomla diz procurar atingir o nível WAI-AA, enquanto que o Plone diz ter certeza de estar enquadrado no nível WAI-AA, mas que alguns pontos do WCAG são subjetivos e que esta interpretação pode variar. Já o Drupal, não possui qualquer recurso de acessibilidade nativamente ou algum documento que indique que tais recursos serão implementados.

Os demais recursos de acessibilidade como o aumento de fonte, alto contraste, leitura de tela e teclas de atalhos são mostradas no Quadro 5.2.

	Drupal	Joomla	Plone
XHTML/CSS	Sim	Sim	Sim
Gera XHTML através do editor visual	Não	Apenas com o <i>plug-in</i> XStandard Lite	Sim
URLs amigáveis	Sim (mod_rewrite)	Sim (mod_rewrite)	Sim (nativo)
Dublin Core	Não	Não	Sim
Aumentar fonte	Não	Sim	Sim
Alto contraste	Não	Sim	Não
Otimizado para leitor de tela	Não informado	Não informado	Sim
Atalhos de Teclado	Não	Não	Sim
WCAG	Não informado	Procura atingir WAI-AA	Diz estar de acordo com o WAI-AA

Quadro 5.2 - Comparação dos padrões Web e Acessibilidade

5.4 Workflow

Tanto o Drupal como o Plone possui um controle avançado de usuários, gerenciado através do uso de grupos. Já no Joomla, o controle de usuários é feito através de sete grupos de permissões fixos, que não podem ser alterados sem a utilização de *plug-ins* pagos.

O Drupal possui suporte a *workflow* através de um *plug-in* que pode ser obtido no seu sítio de extensões. No Plone o *plug-in* de *workflow* já vem em sua instalação, bastando habilitá-lo. O Joomla possui um *plug-in* de *workflow* que funcionava em sua versão 1.0, mas que não está atualizado para funcionar em sua versão atual (Quadro 5.3).

	Drupal	Joomla	Plone
Controle de Usuários	Grupos	Fixo em 7 grupos. <i>Plug-ins</i> pagos adicionam edição de grupos	Usuários e Grupos
Workflow	Com <i>plug-in</i> gratuito	Com <i>plug-in</i> gratuito (mas não funciona na versão atual)	<i>Plug-in</i> nativo que acompanha a instalação

Quadro 5.3 - Comparação dos recursos de Workflow

5.5 Internacionalização e Localização

Todos os CMS analisados possuem suporte ao idioma português e localização brasileira. O Plone vem com todos os idiomas suportados em sua instalação, enquanto nos outros CMS os pacotes de idiomas devem ser baixados separadamente (Quadro 5.4).

	Drupal	Joomla	Plone
Disponível em português	Sim	Sim	Sim
Idiomas	Mais de 40	51	35
Localização	Sim	Sim	Sim

Quadro 5.4 - Comparação da internacionalização e localização

5.6 Facilidade de uso

Dos CMS analisados, apenas o Drupal não possui editor visual WYSIWYG padrão instalado. Em todos os CMS o editor padrão pode ser alterado através do uso de *plug-ins*. Todos eles permitem a formatação, inserção de *links* e *upload* de arquivos (Quadro 5.5).

Do ponto de vista do administrador do sítio, a configuração e personalização do sítio são mais difíceis de serem feitas no Plone, pois exige que se tenha conhecimentos do *Zope*, e sua estrutura de camadas e objetos e a configuração de vários arquivos XML. No Drupal a configuração também não é trivial, pois mesmo os seus recursos básicos de gestão de conteúdos, são feitos através de *plug-ins*, muitos deles exercem a mesma função e o

administrador deve pesquisar e testar vários deles, até encontrar o melhor. Dos três CMS analisados, o Joomla é o mais fácil de configurar e personalizar, pois possui a maior parte dos recursos de gestão de conteúdo é nativa e a instalação de *plug-ins* e configurações são feitas pelo navegador.

	Drupal	Joomla	Plone
WYSIWYG padrão	Apenas com <i>plug-in</i>	TinyMCE	Kupu
Formatação de texto	Com <i>plug-in</i>	Sim	Sim, estilos definidos pelo administrador
Inserção de link	Com <i>plug-in</i>	Sim	Sim, com controle de integridade de <i>links</i> internos
Redimensionador de Imagens	Com <i>plug-in</i>	Sim	Sim
Upload de arquivos	Sim	Sim	Sim
Configuração e personalização	Médio	Fácil	Difícil

Quadro 5.5 - Comparação da facilidade de uso

5.7 Plugins

Todos os CMS analisados possuem suporte para *plug-ins*. A instalação de *plug-ins* no Joomla é a mais fácil, pois basta baixar o pacote no sítio e fazer o *upload* do *plug-in* pelo próprio navegador. No Drupal para instalar *plug-ins* deve se descompactar o pacote no diretório *plug-ins* e habilitá-lo na interface de administração. No Plone, é possível instalar *plug-ins* da mesma maneira que o Drupal, ou através da utilização de Python *eggs*, que possui a vantagem de possuir controle de dependência e a desvantagem de se exigir acesso *shell* a máquina do servidor.

	Drupal	Joomla	Plone
Instalação	Copiar para o diretório	Instalação pelo próprio navegador.	Copiar para o diretório, Utilitário <i>easy_install</i> Utilitário <i>buildout</i>
Controle de dependências	Não	Não	Apenas usando Python <i>eggs</i> .

Quadro 5.6 - Comparação do uso de *plug-ins*

5.8 Segurança

Conforme os relatórios de segurança da Secunia, o Joomla foi o CMS analisado no qual foram encontrados mais falhas de segurança, com 66 no total. No Drupal foram encontradas 22 falhas e no Plone, apenas 4 falhas (Quadro 5.7).

	Drupal	Joomla	Plone
Alertas de segurança	10	19	3
Falhas descobertas	22	66	4
Falhas altamente críticas	0%	0%	33%
Falhas moderadas	40%	63%	33%
Falhas menores	60%	32%	33%
Falhas não críticas	0%	5%	0%

Quadro 5.7 - Comparação da segurança

5.9 Documentação e suporte

Por se tratarem de softwares livres, a maior parte da documentação e suporte dos três CMS é feita pela comunidade de usuários. Apesar do Joomla não possuir oficialmente uma lista de e-mails ou suporte via *chat*, ele possui uma comunidade maior, que faz com seus fóruns de discussão sejam mais ativos. Além disso, o Joomla possui um número maior de publicações impressas, sendo o único com um livro disponível em língua portuguesa (Quadro 5.8).

	Drupal	Joomla	Plone
Documentação	On-line	On-line	On-line
Lista de e-mail	Sim	Não	Sim
Fóruns	Sim	Sim	Sim
Chat	Sim	Não	Sim
Livros publicados	20	26	12
Livros em português	Nenhum	1	Nenhum

Quadro 5.8 - Comparação da documentação e suporte

5.10 Decisão

Decidiu-se pela escolha do Plone como CMS a ser adotado, pois o Plone se mostrou melhor na maioria dos quesitos analisados. O Plone possui recursos de acessibilidade, exigência da lei brasileira para os sítios de instituições públicas, além de estar de acordo com os padrões *Web*, utilizando XHTML e CSS, recomendações do padrão e-gov para desenvolvimento de sítios. O seu editor é fácil de usar e ao mesmo tempo garante que as formatações feitas pelo usuário não quebram as especificações de XHTML do sítio. A sua gerência de permissões pode ser feita de maneira granular, com suporte a *workflow*, tanto globalmente pelo tipo de conteúdo, como pela localização do conteúdo na árvore de navegação do sítio. Além disso, possui interface em língua portuguesa, suporta a instalação de *plug-ins*, com controle de dependências e possui um histórico de poucas falhas de segurança encontradas. Um dos seus pontos fracos é a exigência de um grande tempo de aprendizado para o administrador entender sua estrutura de camadas, objetos e arquivos de configurações, além de exigências técnicas diferentes do tradicional LAMP (Linux, Apache, MySQL e PHP).

No caso do Drupal, sua principal deficiência é a falta de recursos de acessibilidade e carência de um editor WYSIWYG que garanta a manutenção de um código XHTML válido.

O Joomla possui a desvantagem de ter fixo os grupos de usuários, obrigando a utilização das suas permissões padrões, que podem não refletir as necessidades reais do utilizador. Além disso, os requisitos de permissões de uma organização são dinâmicos e podem mudar com o tempo. A falta de um *plug-in* de *worklow* que funcione na versão do Joomla analisada também é um ponto fraco, pois centraliza a publicação de conteúdos no sítio.

Capítulo 6

Conclusão

Conforme uma organização cresce, o volume de informações por ela gerado também cresce, aumentando a demanda para divulgá-las na internet. O uso de um sistema de gestão de conteúdos permite agilizar o processo de publicação de conteúdo ao descentralizá-lo, permitindo que usuários não técnicos, no papel de autores, sejam capazes de publicarem páginas na *web*.

Este trabalho avaliou as principais ferramentas CMS de código livre, focando-se nas principais necessidades do estudo de caso, UTFPR - Campus Francisco Beltrão e nas recomendações do padrão e-gov para desenvolvimento de sítios de instituições do governo federal.

De acordo com os critérios avaliados, conclui-se que o Plone é o mais indicado para a implantação do sistema. Apesar das exigências técnicas diferenciadas e de exigir uma grande curva de aprendizagem, o Plone se mostrou melhor na maioria dos itens analisados.

Um parâmetro recomendado pelo padrão e-gov para a avaliação de um CMS que não foi considerado neste trabalho é a questão do desempenho. A medição do desempenho dos CMS, exigiria um estudo mais detalhado com a escolha das técnicas e métricas apropriadas, a utilização de um ambiente controlado, entre outras variáveis. Preferiu-se não incluir este critério na comparação dos CMS sob pena de se ter resultados imprecisos ou incorretos. Tal comparação poderá ser feita em um trabalho futuro.

Referências Bibliográficas

BOIKO, B. *Content management bible, 2nd edition*. 2nd. ed. Indianapolis, Ind.: Wiley Pub., 2005.

BOISOT, M. *Knowledge assets : securing competitive advantage in the information economy*. Oxford ; New York: Oxford University Press, 1998.

CONTENTMANAGER. What is a content management system? , 2008. Disponível em:<<http://www.contentmanager.eu.com/cms.htm>>. Acesso em: 12/10/2008.

CRUZ, T. *Gerência do conhecimento*. São Paulo: Ed. Cobra, 2002.

Dublin Core Metadata Initiative. 2008. Disponível em:<<http://dublincore.org/>>. Acesso em: 06/10/2008.

GOOGLE. *Google Insights for Search*. 2008a. Disponível em:<<http://www.google.com/insights/search/#cat=13&q=mambo%2Cjoomla&cmpt=q>>.

GOOGLE. Razões para se usar o Google. 2008b. Disponível em:<http://www.google.com.br/why_use.html>. Acesso em: 12/11/2008.

IFLA. Digital Libraries: Metadata Resources. 2005. Disponível em:<<http://www.ifla.org/II/metadata.htm>>. Acesso em: 12/11/2008.

INMON, W. H. *et al. Business metadata : capturing enterprise knowledge*. Elsevier/Morgan Kaufmann, 2008. (Amsterdam; Boston).

JOOMLA. Joomla - Accessibility Statement. 2008. Disponível em:<<http://help.joomla.org/content/view/805/125/>>.

MCGEE, J.; PRUSAK, L. *Gerenciamento estratégico da informação*. Rio de Janeiro: Campus, 1994.

MCKAY, A. *The Definitive Guide to Plone*. Apress, 2004.

MCNAY, H. E. Enterprise content management: an overview. *Professional Communication Conference, 2002. IPCC 2002. Proceedings. IEEE International [S.I.]*, 2002.

OLIVEIRA, D. P. R. *Sistemas de informações gerenciais: estratégias, táticas, operacionais*. 4. ed. São Paulo: Atlas, 1997.

Padrões Brasil e-GOV - Governo Eletrônico. 2008. Disponível em: <<http://www.governoeletronico.gov.br/acoes-e-projetos/padroes-brasil-e-gov>>. Acesso em: 20/10/2008.

PEREIRA, J. C. L.; BAX, M. P. Introdução à Gestão de Conteúdos. In: *Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento*, 2002. 2002.

PHP-NUKE. PHP-Nuke Commercial License. 2008. Disponível em: <http://phpnuke.org/modules.php?name=Commercial_License>. Acesso em: 10/11/2008.

PLONE. Existing features — Plone CMS: Open Source Content Management. 2008. Disponível em: <<http://plone.org/products/plone/features/3.0/existing-features/referencemanual-all-pages>>. Acesso em: 09/11/2008.

SECUNIA. Secunia.com. 2008. Disponível em: <<http://www.secunia.com/>>. Acesso em: 12/11/2008.

WFMC. The Workflow Reference Model. 1995. Disponível em: <<http://www.wfmc.org/>>. Acesso em: 12/10/2008.

Apêndice A

Glossário

Backend: No contexto de CMSs, backend é a interface de administração do site.

Blog: É um sistema de publicação na *web* destinado a divulgar informação por ordem cronológica, à semelhança de um diário.

FAQ: Acrônimo de *Frequently Asked Questions*, é uma seção com respostas para perguntas freqüentes nos sites.

Frontend: Interface de interação entre o sistema e o usuário.

Proxy: Servidor que atende a requisições repassando os dados a outros servidores.

Root: Usuário que possui acesso a todos os recursos do sistema, sem restrições, utilizado para administração de sistemas.

Shell: Programa que recebe, interpreta e executa os comandos de usuário, aparecendo na tela como uma linha de comandos, representada por um prompt, que aguarda os comandos do usuário

Syndication: tornar o material de um site disponível a sua audiência ou a outros sites, provendo um resumo do conteúdo adicionado recentemente. Exemplos são *feeds* RSS ou Atom.

W3C: Acrônimo de *World Wide Web Consortium*, é um consórcio mundial fundado em 1994, responsável pela criação de padrões que regem a *Web*.

Wiki: software colaborativo permite a edição coletiva dos documentos usando um sistema que não necessita que o conteúdo tenha que ser revisto antes da sua publicação.

Apêndice B

Lista de livros publicadas sobre os CMS

Drupal

Build Websites With Drupal, 100 Most Asked Questions on Drupal - The Free and Open Source modular framework and Content Management System (CMS) por Andrew Haney (2008)

Building Online Communities With Drupal, phpBB, and WordPress por Robert T. Douglass, Mike Little, e Jared W. Smith (2005)

Building powerful and robust websites with Drupal 6 por David Mercer (2008)

Cracking Drupal: A Drop in the Bucket por Greg Knaddison (2009)

Drupal 5 Themes por Ric Shreves (2007)

Drupal 6 por Thorsten P. Luhm (2008)

Drupal 6 Themes por Ric Shreves (2008)

Drupal A Beginner's Guide por Ray West (2009)

Drupal Essential Training por Tom Geller (versão em cd-rom, 2008)

Drupal Multimedia por Aaron Winborn (2008)

Drupal: Creating Blogs, Forums, Portals, and Community Websites por David Mercer (2006)

Front End Drupal: Designing, Theming, Scripting (Developer's Library) por Konstantin Käfer and Emma Hogbin (2009)

Learning Drupal 6 Module Development por Matt Butcher (2008)

Leveraging Drupal: Getting Your Site Done Right por Victor Kane (2009)

Practical Drupal: Evaluating and Using a Web Content Management System por Niall

Mansfield (2008)

Pro Drupal Development por John K. VanDyk e Matt Westgate (2007)

Pro Drupal Development, Second Edition (Beginning from Novice to Professional) por John K. VanDyk. (2008)

Professional Drupal por Trevor Twining (2009)

Selling Online with Drupal e-Commerce (From Technologies to Solutions) por Michael Peacock (2008)

Using Drupal por Angela Porron, Heather Berry, Nathan Haug, and Jeff Eaton (2008)

Joomla

Joomla!: Guia do Operador por BARRIE NORTH (2008)

Advanced Joomla! por Dan Rahmel (2009)

Beginning Joomla!: From Novice to Professional, Second Edition (Beginning from Novice to Professional) por Dan Rahmel (2009)

Building Web 2.0 Business Websites: Business Process Innovation with Web 2.0 and Joomla! por Jackley Cesar (2008)

Building Websites with Joomla! 1.5: The best-selling Joomla! tutorial guide updated for the latest 1.5 release por Hagen Graf (2008)

Dodging the Bullets: A Disaster Preparation Guide for Joomla!™ Web Sites por Thomas Canavan (2007)

Fundamentals of Joomla! (Video Training) por Barrie M. North (2009)

Joomla por Florian Harms (2006)

Joomla! 1.5 & Virtuemart Websites for Small Business! por PhD Michelle M. Griffin (2008)

Joomla! 1.5 & Virtuemart Websites for Sports and Clubs por Michelle M. Griffin (2008)

Joomla! 1.5 & VirtueMart: The Power is at your Fingertips! por Michelle M. Griffin PhD (2008)

Joomla! 1.5 por Hagen Graf (2008)

Joomla! A User's Guide: Building a Successful Joomla! Powered Website por Barrie M. North (2007)

Joomla! Accessibility: A quick guide to creating accessible websites with Joomla! por Joshue O Connor (2007)

Joomla! Cash: Money-making weapons for your Joomla! website por Brandon Dawson (2007)

Joomla! For Dummies (For Dummies (Computer/Tech)) por Steve, Ph.D. Holzner e Nancy Conner (2009)

Joomla! Template Design: Create your own professional-quality templates with this fast, friendly guide: A complete guide for web designers to all aspects... Joomla! 1.0.8 PHP Content Management System por Tessa Blakeley Silver (2007)

Joomla! Templates: The Official Guide por Chris Davenport (2009)

Joomla! Web Security por Tom Canavan (2008)

Joomla!: Visual QuickStart Guide por Marni Derr and Tanya Symes (2008)

Learning Joomla! 1.5 Extension Development: Creating Modules, Components, and Plugins with PHP por Joseph L. LeBlanc (2007)

Mastering Joomla! 1.5 Extension and Framework Development: The Professional Guide to Programming Joomla! por James Kennard (2008)

Open Source Pro: Joomla por Tim, Jowers (2007)

Professional Joomla! (Programmer to Programmer) por Dan Rahmel (2007)

The Definitive Guide to Joomla por Mitchell Pirtle (2005)

The Joomla Admin Manual: A Step por Step Guide to a Successful Website por Barrie North (2007)

Website Design Guide to Joomla! 1.5, Virtuemart & Extensions por PhD (abd), Michelle M. Griffin (2008)

Plone

A User's Guide to Plone: Updated for Plone 3 por Thomas Lotze, Alan Runyan, Jan Ulrich Hasecke, e Robert Nagle (2008)

Building Websites with Plone: An in-depth and comprehensive guide to the Plone content management system. por Cameron Cooper (2004)

Content Management with Plone: Handbook for Authors and Editors: Updated for Plone 2 por Thomas Lotze and Christian Theune (2006)

Plone 2.0 por Peter Walerowski (2005)

Plone por Andy McKay (2005)

Plone Content Management Essentials por Julie C. Meloni (2004)

Plone Live por Michel; Shariff, Munwar Pelletier (2005)

PLONE: CONTENT MANAGEMENT ESSENTIALS por Julie C. Meloni (2005)

Practical Plone 3: A Beginner's Guide to Building Powerful Websites por Sam Knox, Jon Stahl, Martin Aspeli, e David Convent (2009)

Professional Plone Development por Martin Aspeli (2007)

The Definitive Guide to Plone por Andy McKay (2004)

Web Component Development with Zope 3 por Philipp von Weitershausen e P. J. Eby (2008)

Maringá, 24 de março de 2009.

À Coordenação de Trabalho de Especialização
Departamento de Informática
Centro de Tecnologia
Universidade Estadual de Maringá

Encaminhamos, em anexo, a monografia intitulada “UM ESTUDO COMPARATIVO DE SISTEMAS DE GESTÃO DE CONTEÚDO PARA A MANUTENÇÃO DE SÍTIOS WEB EM UNIVERSIDADES PÚBLICAS BRASILEIRAS” para avaliação desta coordenação.

Aluno: Leandro Koiti Sato

Orientador: Prof. Dante Alves Medeiros Filho