



Centro de Competência de Ciências Exactas e da Engenharia

# **Estudo Comparativo de CMSs**

## **Estudo de Caso: uma IPSS**

Dissertação de Mestrado

Mónica Raquel Pereira Baptista

Orientador(a): Karolina Baras

Novembro 2014



## ÍNDICE

Índice de Figuras.....	v
Índice de Tabelas .....	v
Resumo .....	vii
<i>Abstract</i> .....	ix
Acrónimos.....	xi
1. Introdução.....	13
1.1. Contextualização.....	13
1.2. Objectivos .....	15
1.3. Estrutura da tese.....	15
2. Estado da Arte.....	17
2.1. O que é um CMS .....	17
2.2. Joomla.....	18
2.2.1. Interface de administração.....	19
2.2.2. Estrutura interna do Joomla.....	20
2.2.3. Processo de actualização do CMS .....	21
2.2.4. Construção de <i>Plugins</i> .....	21
2.3. Drupal.....	24
2.3.1. Interface de administração.....	25
2.3.2. Estrutura interna do Drupal .....	25
2.3.3. Processo de actualização do CMS .....	26
2.3.4. Construção de <i>Plugins</i> .....	27
2.4. WordPress.....	29
2.4.1. Interface de administração.....	30
2.4.2. Estrutura interna do WordPress.....	31
2.4.3. Processo de actualização do CMS .....	32
2.5. Comparação dos CMS.....	33
2.5.1. O mais pesquisado.....	33
2.5.2. Complexidade.....	34

2.5.3.	Recursos.....	34
2.5.4.	Popularidade.....	35
2.5.5.	Vantagens e Desvantagens.....	36
2.5.5.1.	Drupal.....	36
2.5.5.2.	Joomla.....	36
2.5.5.3.	WordPress.....	37
2.5.6.	Ficha Técnica.....	39
2.6.	Escolha dos CMS.....	39
2.6.1.	Versões dos CMS.....	40
2.6.1.1.	Drupal 7.....	40
2.6.1.2.	WordPress 3.6.....	41
2.6.1.3.	Joomla 3.x.....	40
2.6.2.	Realização de tarefas.....	41
2.6.2.1.	Criar um utilizador.....	41
2.6.2.2.	Criar uma página.....	43
2.6.2.3.	Criar artigos.....	44
2.6.2.4.	Instalar Plugins.....	44
2.6.2.5.	Criar banner.....	45
3.	Desenvolvimento.....	47
3.1.	Lista de requisitos.....	47
3.1.1.	Requisitos Funcionais.....	47
3.1.1.1.	Do lado do sistema.....	47
3.1.1.2.	Do lado do site.....	48
3.1.2.	Requisitos Não-Funcionais.....	48
3.2.	Soluções propostas.....	48
3.2.1.	Formulários.....	49
3.2.1.1.	Joomla.....	49
3.2.1.2.	Drupal.....	51
3.2.2.	Loja Virtual.....	52
3.2.2.1.	Joomla.....	52

3.2.2.2.	Drupal .....	55
3.2.3.	Calendário e Eventos .....	57
3.2.3.1.	Joomla .....	57
3.2.3.2.	Drupal .....	57
3.3.	<i>Website</i> .....	58
3.3.1.	Configurando o SMTP .....	60
4.	Discussão .....	61
5.	Conclusão .....	65
	Trabalhos Futuros .....	66
	Referências .....	67
	Anexos .....	73
Anexo 1	Diagrama Entidade-Relação do Drupal (Parte 1) .....	74
Anexo 1	Diagrama Entidade-Relação do Drupal (Parte 2) .....	75
Anexo 1	Diagrama Entidade-Relação do Drupal (Parte 3) .....	76
Anexo 2	Diagrama Entidade-Relação do WordPress.....	77



## ÍNDICE DE FIGURAS

Figura 1 – Anatomia dos CMS.....	14
Figura 2 – Interface de administrador do Joomla .....	19
Figura 3 – Código XML do ficheiro de instalação do <i>plugin</i> .....	22
Figura 4 – Código PHP da função do <i>plugin</i> .....	24
Figura 5 – Interface de administrador do Drupal .....	25
Figura 6 – Estrutura do ficheiro <i>.info</i> .....	27
Figura 7 – Opção adicional no módulo <i>Path</i> .....	27
Figura 8 – Conteúdo do ficheiro <i>person.info</i> .....	27
Figura 9 – Resultado do ficheiro <i>person.info</i> .....	28
Figura 10 – Código PHP do ficheiro <i>person.module</i> .....	28
Figura 11 – Código da função de retorno.....	29
Figura 12 – Visualização da funcionalidade do <i>plugin</i> no <i>front-end</i> do <i>site</i> .....	29
Figura 13 – Interface de administrador do WordPress .....	30
Figura 14 – Número de Pesquisas no <i>Google</i> .....	34
Figura 15 – Comparação de recursos/características .....	34
Figura 16 – CMS preferido dos utilizadores .....	35
Figura 17 – O CMS mais utilizado .....	35
Figura 18 – <i>HomePage</i> do <i>website</i> em Joomla.....	59
Figura 19 – <i>Homepage</i> do <i>website</i> em Drupal .....	60

## ÍNDICE DE TABELAS

Tabela 1 - Descrição das tabelas da base de dados do Joomla 3.1 .....	20
Tabela 2 - Descrição das tabelas do Diagrama E-R.....	32
Tabela 3 - Breve Comparação entre o Joomla, Drupal e WordPress .....	38
Tabela 4 - Ficha técnica dos CMS líderes.....	39



## RESUMO

Sistemas de Gestão de Conteúdo (*Content Management System* - CMS) são sistemas que têm como objectivo facilitar o processo de desenvolvimento de *websites* pessoais e de pequenas empresas através de diversas funcionalidades não exigindo aos utilizadores conhecimento na área da programação, deixando de haver a necessidade de recorrer a empresas de terceiros para a execução do mesmo.

Esta dissertação tem por objectivo o estudo de três CMSs – quanto às suas características individuais, popularidade, complexidade, flexibilidade, escalabilidade, qualidade do produto final, entre outros aspectos – e o desenvolvimento de um *website* para a instituição particular de solidariedade social Centro da Mãe. Este *website* irá ser desenvolvido em dois dos CMSs estudados, que serão escolhidos por meio do estudo de cada um deles, o que as suas comunidades oferecem e experiência pessoal. E desta forma poder compará-los.

**Palavras-chave:** CMS, Drupal, Joomla, WordPress.



## **ABSTRACT**

*Content Management Systems (Content Management System – CMS) aim to facilitate the process of developing personal websites and small businesses through several features not requiring users knowledge in the area of programming, leaving out the need to resort to third-party companies for execution.*

*This thesis aims to study three CMSs – as to their individual characteristics, popularity, complexity, flexibility, scalability, quality of the final product, among others aspects – and to develop a website for the institution called Centro da Mãe. This website will be developed in two of the CMSs studied, selected based on the study of each of them, of what their communities offer and personal experience. A comparison between the two will be presented.*

**Keywords:** CMS, Drupal, Joomla, WordPress.



## ACRÓNIMOS

API – *Application Programming Interface*

BD – Base de Dados

CMS – *Content Management System*

CSS – *Cascading Style Sheets*

EUA – Estados Unidos da América

HTML – *HyperText Markup Language*

IE – *Internet Explorer*

IPSS – Instituições Particulares de Solidariedade Social

PHP – “**PHP: Hypertext Preprocessor**”, originalmente **Personal Home Page**

PME – Pequenas e Médias Empresas

SEM – *Search Engine Marketing*

SEO – *Search Engine Optimization*

URL – *Uniform Resource Locator*

WYSIWYG – **What You See Is What You Get**

XML – *eXtensible Markup Language*



## 1. INTRODUÇÃO

Antes da chegada dos sistemas de gestão de conteúdos, para inserir algum texto na internet era preciso aprender alguma linguagem de marcação para a *web* como o HTML ou então colocar os artigos em texto simples. O tempo avançou, e algumas ferramentas foram criadas com o intuito de simplificar esse trabalho, criando a formatação a partir do texto inserido, e assim, a criação de *websites* e a *internet* foram-se popularizando como um todo. Com isto, as necessidades mudaram, ficheiros em texto simples tornaram-se aborrecidos de se ler, bem como a manutenção dos *sites*, então, para resolver algumas dessas questões foram criados os Sistemas de Gestão de Conteúdos, mais conhecidos como CMS (*Content System Management*). [1]

Os primeiros CMS tinham como principais características um ambiente de desenvolvimento muito estruturado em que o utilizador tinha de usar *tags* e *templates*, pois não havia editores *WYSIWYG*<sup>1</sup> – *What You See Is What You Get*. Os utilizadores tinham de ter alguns conhecimentos de HTML, caso contrário não estariam habilitados para editar o *site*.

Na segunda fase de desenvolvimento dos CMS, foram construídas e implementadas as características de *WYSIWYG*, pesquisa e melhoramentos no HTML. Durante essa fase, começou o movimento *open source* para os CMSs, dando origem à divisão entre aplicações pagas e gratuitas. Com isto, os CMS tornaram-se ricos em recursos e com o passar dos anos – até aos dias de hoje – o número de CMS disponíveis foi crescendo. [2]

Assim, o intuito deste projecto foca-se no estudo de três Sistemas de Gestão de Conteúdo – Joomla, Drupal e WordPress – de modo a conhecer e compreender o funcionamento de cada um, enquanto utilizador destes sistemas, e as funcionalidades adicionais que oferecem, implementando com esse objectivo, um *website* para uma IPSS (Instituição Particular de Solidariedade Social).

O *website* desenvolvido direcciona-se para a IPSS Centro da Mãe, cujo propósito é o de apoiar grávidas adolescentes, jovens mães e os seus filhos. Este *website*, visa divulgar a informação referente ao Centro da Mãe, de modo a alcançar um maior número de pessoas, e ainda disponibilizar uma loja *online* em que as vendas revertem a favor da instituição.

### 1.1. CONTEXTUALIZAÇÃO

A utilização de *software Open Source* tem ganho maior ênfase ultimamente devido à necessidade cada vez maior de diminuir os encargos relacionados com o *software* proprietário. Assim, pretende-se com esta tese estudar a fundo os Sistemas de Gestão de

---

<sup>1</sup> **WYSIWYG** – termo usado para classificar ferramentas de edição e desenvolvimento que permitem visualizar, em tempo real, exactamente aquilo que será publicado ou impresso.

Conteúdo (CMS) mais utilizados, perceber as suas características, vantagens e desvantagens e avaliar o custo de adaptação de cada um deles a uma situação concreta, no caso uma IPSS.

Os CMSs são Sistemas de Gestão de Conteúdo – de *websites*, portais e *intranets*<sup>2</sup> – que permitem a introdução e gestão de conteúdo em tempo real de forma rápida e dinâmica, sem a necessidade de programação. Quando alocado num servidor *online*, o administrador pode aceder ao *site* a partir de qualquer computador ou equipamento com ligação à internet e gerir não só o conteúdo como também funcionalidades administrativas, como por exemplo, gerir a estrutura do *site*, a aparência das páginas publicadas, entre outras.

Estes sistemas oferecem métodos para estender ou adicionar funcionalidades através do uso de *plugins*, módulos ou extensões, sem qualquer esforço por parte do utilizador. Também disponibilizam *templates* que determinam a aparência do *site*, bem como a sua estrutura consoante o *template*.

O CMS gere todo o ciclo de vida das páginas de um *site*, desde a criação ao armazenamento. As suas funcionalidades básicas são divididas em quatro categorias, como ilustra a **Figura 1**.

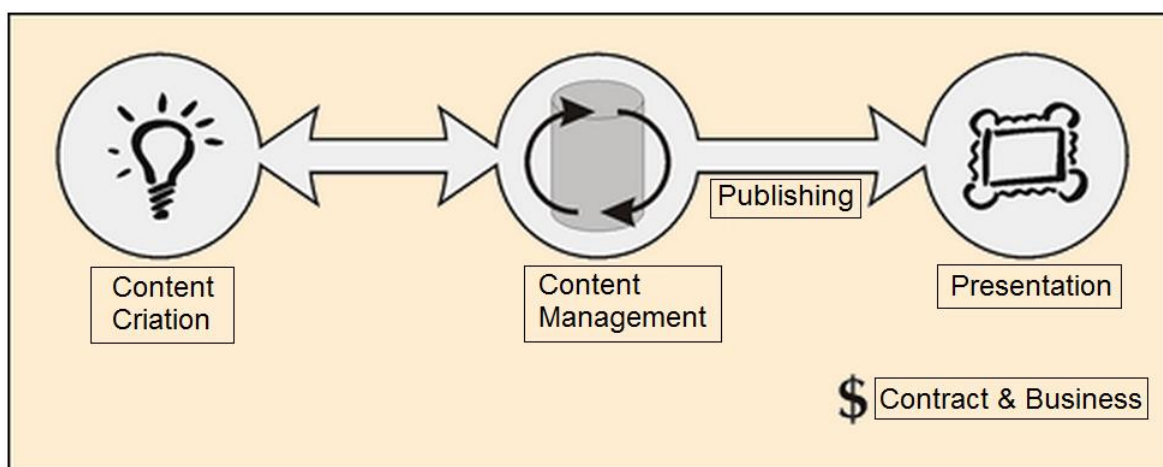


Figura 1 - Anatomia dos CMS [3]

A primeira categoria é a **criação de conteúdo**, que é o ambiente de criação projectado para funcionar como um editor de texto. Este ambiente é o modo de criar novas páginas de forma fácil e/ou actualizar conteúdo sem precisar saber HTML.

Uma vez que uma página tenha sido criada, esta é guardada no repositório central – base de dados – do CMS juntamente com todo o conteúdo do *site* e outros detalhes. Através da informação guardada na base de dados é possível obter uma série de recursos úteis de

<sup>2</sup> **Intranets** – rede de computadores pública ou privada, de uso exclusivo de um determinado local, como por exemplo, uma empresa, que só pode ser acedida pelos utilizadores internos para partilha de informações restritas.

**gestão de conteúdo**, como controlar todas as versões de uma página, quem a modificou e quando, garantir que cada utilizador só tem acesso à informação que lhe é permitida, etc.

Depois do conteúdo estar armazenado no repositório este pode ser **publicado** em um ou mais *sites*. O CMS automatiza totalmente a publicação de um *site* garantindo a consistência das páginas em todo o *site*.

Por fim, o CMS pode também fornecer alguns recursos para melhorar a qualidade e a eficácia do próprio *site*, como por exemplo, a construção da navegação do *site*, através da leitura da estrutura directamente da base de dados.

[3]

## **1.2. OBJECTIVOS**

Os objectivos principais desta dissertação são o estudo dos CMS propostos, e o desenvolvimento de um *website* em pelo menos dois deles.

A abordagem utilizada para realizar os objectivos foi a análise de cada CMS, identificação das suas características e comparação dos CMS quanto à sua popularidade, complexidade, flexibilidade, escalabilidade e qualidade do produto final, entre outros aspectos.

## **1.3. ESTRUTURA DA TESE**

Esta tese está organizada em quatro capítulos, além da introdução, que serão descritos de seguida:

- 1. Introdução** – este capítulo contém a contextualização, objectivos e estrutura da tese.
- 2. Estado da Arte** – neste capítulo é descrito o que é um CMS, os três CMS a serem estudados, bem como a comparação entre eles e o guia de realização de tarefas importantes em cada CMS.
- 3. Desenvolvimento** – inclui a lista de requisitos do *website* desenvolvido e as soluções propostas para executar tarefas além das existentes por defeito no CMS.
- 4. Discussão** – neste capítulo são descritos os benefícios e os custos associados à utilização de um CMS para uma IPSS.
- 5. Conclusão** – neste capítulo são apresentadas as conclusões finais.



## 2. ESTADO DA ARTE

Este capítulo é dedicado à apresentação geral de Sistemas de Gestão de Conteúdos e dos três mais apreciados pelos utilizadores. Posteriormente à descrição dos CMS a estudar, é feita uma comparação entre eles, desde a sua popularidade, complexidade, recursos, entre outros aspectos, à escolha dos dois CMS para o desenvolvimento do *website*.

### 2.1. O QUE É UM CMS

Sistemas de Gestão de Conteúdo (CMS) são sistemas/aplicações que permitem a publicação, edição e/ou alteração de conteúdo, bem como a manutenção de um *site* a partir de uma página central. Estes sistemas armazenam o conteúdo real, que pode ser texto ou multimédia, numa base de dados.

Criados com o intuito de ajudar os utilizadores a publicar documentos e/ou ficheiros de multimédia de uma forma mais fácil, isto é, com menos intervenções técnicas, e de uma forma mais consistente e automatizada, estes sistemas vieram a acelerar o processo de actualização e a abranger um maior número de utilizadores. [4]

Segue-se uma lista de benefícios dos CMS: [5]

- ✓ **Gestão de página rápida e fácil** – qualquer utilizador autorizado pode rápida e facilmente publicar *online* sem *software* complicado ou conhecimentos de programação;
- ✓ **Gestão de fluxo de trabalho** – um processo de fluxo de trabalho integrado possibilita melhor gestão de conteúdo;
- ✓ **Flexibilidade para os programadores** – permite aos utilizadores não-técnicos publicar conteúdo facilmente, e aos programadores, concentrarem-se nas funcionalidades e recursos avançados;
- ✓ **Design é separado do conteúdo** – é possível manipular o conteúdo, sem medo de acidentalmente alterar o *design*;
- ✓ **Orientado à base de dados** – o utilizador só precisa alterar os dados uma vez para que possam ser actualizados em todo o *site*;
- ✓ **Recursos partilhados** – os gestores do *site* terão acesso a recursos partilhados, tais como, módulos, imagens, ficheiros de áudio e vídeo, etc.;
- ✓ **Sistemas de aprovação** – o administrador pode atribuir diferentes níveis de acesso a diferentes utilizadores, e o CMS tem mecanismos para garantir que o conteúdo seja aprovado antes de ser publicado;
- ✓ **Preparado para ambiente móvel** – o CMS dimensiona automaticamente o *site* para se ajustar a *tablets* e dispositivos móveis;

- ✓ Capacidades de arquivo – o administrador pode rastrear quem fez alterações nas páginas e arquivar versões anteriores das mesmas;
- ✓ Acesso remoto – o administrador pode aceder e actualizar o *site* de qualquer lugar com conexão à *internet*;
- ✓ Segurança – a segurança é automática;
- ✓ Motor de pesquisa amigável – o CMS ajuda a otimizar o *site* para que os utilizadores do motor de busca possam encontrar facilmente a informação;
- ✓ Actualizações – o CMS permite configurar alertas para notificar o editor quando o conteúdo precisa ser revisto, actualizado ou removido. Isto ajuda a evitar que os dados antigos sejam apresentados, mostrando informação desactualizada aos utilizadores.

Como em todos os sistemas, os CMS também têm alguns inconvenientes e trabalhar com um “modelo” pré-concebido, pode ser uma desvantagem para um utilizador sem conhecimentos em programação, pois implica aceitar algumas características, restrições ou limitações, como a aparência do *site* limitada pelo sistema de *templates* do CMS e a necessidade de aprendizagem.

Antes de escolher um CMS é necessário considerar algumas questões relevantes [4] [6]:

1. Quanto conhecimento tem no que diz respeito ao desenvolvimento?
2. Quanta expansão espera ter de fazer?
3. Quanta codificação personalizada espera realizar?
4. O *template* desejado está disponível?
5. Quantos níveis diferentes de grupos de utilizador espera ter?
6. Quão intuitiva é a administração do CMS?

De entre os diversos CMSs existentes, segue-se uma descrição mais pormenorizada do Joomla 3.x, Drupal 7 e WordPress 3.6 – CMSs de caso de estudo nesta tese.

## 2.2. JOOMLA

Designado como um dos CMS *Open Source* mais poderoso e completo, o Joomla foi projectado para a criação de *sites* em vários idiomas num curto espaço de tempo, incorpora vários tipos de conteúdo e hierarquias de utilizadores [7] [8]. Com o Joomla é possível criar aplicações para comércio electrónico (*e-commerce*), portais, blogues, entre outros tipos de *sites*. Este CMS pode ser caracterizado como um CMS “universal”, pelo facto de poder ser personalizado como o utilizador quiser, adicionando funcionalidades por meio de *add-ons* (complementos tais como, *plugins*, módulos, etc.).

Algumas das suas principais características são:

- ✓ Gestão de utilizadores e conteúdo multimédia;
- ✓ Gestão de *banners* e *templates*;
- ✓ Gestão de conteúdo e ligações *web*;
- ✓ Pesquisa e sistema de ajuda integrado;
- ✓ Extensibilidade poderosa.

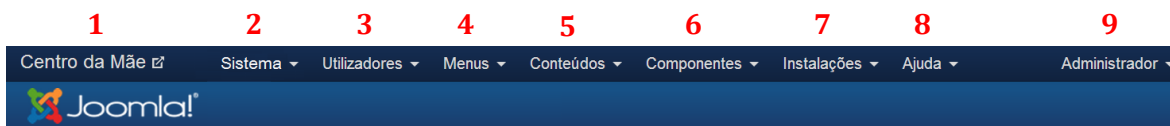
O Joomla tem uma comunidade forte e útil, onde os membros mais activos fornecem respostas rápidas a dúvidas que possam surgir, e ainda programadores que trabalham a fim de tornar este CMS cada vez melhor. Em [8] Stansberry refere que muitos dos *plugins* disponibilizados pela comunidade de desenvolvimento Joomla são pagos, enquanto que as outras comunidades de *CMSs Open Source*, fornecem os *plugins* gratuitamente.

Este sistema é também conhecido pela facilidade de uso, até mesmo para alguém que é principiante, e ainda possui *plugins* para facilitar certas funcionalidades que se deseje introduzir no *site*, como por exemplo, *Google Maps*, calendários, galeria de imagens, carrinho de compras, anúncios de classificados, e muito mais. [7]

Segue-se a apresentação da interface de administração com uma breve legenda elucidativa, e a estrutura interna do Joomla 3.x, nas secções **2.2.1** e **2.2.2**, respectivamente.

### 2.2.1. Interface de administração

A **Figura 2** mostra a interface de administração do Joomla. Para auxiliar a apresentação da interface recorreu-se ao uso da numeração para os menus.



**Figura 2 – Interface de administrador do Joomla**

1. **Centro da Mãe** – Este menu tem o nome atribuído ao *site*, que neste caso é **Centro da Mãe** e direcciona o administrador para o *front-end* do mesmo.
2. **Sistema** – Permite o acesso às configurações de administração, como o painel de controlo, configurações globais, etc.
3. **Utilizadores** – Permite adicionar e gerir contas de utilizadores, grupos de utilizadores e as suas permissões (nível de acesso). Através deste menu também é possível o envio de *email* em massa.
4. **Menus** – Permite criar menus do tipo “grupos de menus” que servem para agrupar os “submenus”. Por exemplo, o menu ‘*user menu*’ agrupa os menus de utilizador (perfil, submeter artigo, etc). Permite também criar os tais “submenus” que são os menus que aparecem no *front-end* do *site*, como por exemplo, **Início** e **Contactos**.
5. **Conteúdos** – Permite criar e gerir artigos e as categorias dos mesmos.

6. Componentes – Permite criar e gerir funcionalidades padrão ou que foram instaladas pelo administrador, como por exemplo, *plugins* e componentes.
7. Instalações – Permite instalar e gerir instalações de módulos, *plugins*, *templates* e idiomas.
8. Ajuda – Fornece ajuda sobre o Joomla e a sua comunidade.
9. Administrador – Permite editar a conta de administrador e encerrar sessão.

### 2.2.2. Estrutura interna do Joomla

A base de dados do Joomla 3 consiste em 64 tabelas das quais 29 serão descritas na **Tabela 1**, pois são as tabelas ainda existentes desde a versão 1.6.

**Tabela 1 - Descrição das tabelas da base de dados do Joomla 3.1 [9]**

Joomla 3.1 – 64 Tabelas	
Nome da tabela	Descrição
kpdu5_assets	Esta tabela é usada sempre que há uma verificação se um utilizador está autorizado a executar uma acção.
kpdu5_banners	Contém a informação de cada <i>banner</i> definido no <i>site</i> .
kpdu5_categories	Armazena cada categoria definida no <i>site</i> . Isto inclui categorias de artigos, <i>banners</i> , contactos, <i>newsfeeds</i> e <i>weblinks</i> .
kpdu5_contact_details	Guarda os dados de cada contacto definido no <i>site</i> .
kpdu5_content	Contém a informação de cada artigo definido no <i>site</i> .
kpdu5_content_frontpage	Contém os dados de artigos atribuídos ao item de menu em destaque.
kpdu5_content_rating	Armazena os dados da classificação dos artigos.
kpdu5_core_log_searches	Guarda os termos de pesquisa registados pelo <i>site</i> .
kpdu5_extensions	Armazena a informação sobre cada extensão instalada no <i>site</i> . Isto inclui extensões de componentes, bibliotecas, módulos, <i>plugins</i> , <i>templates</i> e idiomas.
kpdu5_languages	Esta tabela armazena os idiomas instalados.
kpdu5_menu	Armazena os dados de cada item do menu <i>front-end</i> e <i>back-end</i> definido para o <i>site</i> .
kpdu5_messages	Esta tabela armazena os dados das mensagens privadas enviadas no <i>site</i> .
kpdu5_messages_cfg	Contém uma linha para cada utilizador <i>back-end</i> que define uma configuração na opção <b>Componentes &gt; Mensagens</b> .
kpdu5_modules	Armazena a informação dos módulos definidos no sistema. A coluna " <i>client_id</i> " quando igual a "0" indica que é um módulo <i>front-end</i> e quando igual a "1" indica que é um módulo de <i>back-end</i> .
kpdu5_modules_menu	Esta é uma tabela de mapeamento para mostrar que os módulos são atribuídos a itens de menu. A coluna " <i>menuid</i> " indica que os itens do

	menu do módulo está programado da seguinte forma: “0” significa atribuído a todos os itens do menu; nº inteiro positivo significa atribuído a este item de menu; e nº inteiro negativo significa atribuído a todos os itens de menu, excepto para este item de menu.
kpdu5_newsfeeds	Guarda os dados relativos às <i>newfeeds</i> criadas no <i>site</i> .
kpdu5_redirect_links	Esta tabela armazena os dados de cada redireccionamento criado no <i>site</i> .
kpdu5_schemas	Contém uma linha para cada extensão que fez mudanças na base de dados durante a sua instalação, juntamente com a última versão instalada da extensão.
kpdu5_session	Armazena cada sessão activa do <i>site</i> .
kpdu5_template_styles	Armazena cada estilo de <i>template</i> definido para o <i>site</i> .
kpdu5_updates	Contém uma linha para cada pacote disponível para ser instalado.
kpdu5_update_categories	Esta tabela é usada para categorizar as actualizações. Ela é mantida automaticamente pelo Joomla.
kpdu5_update_sites	Lista de actualização do <i>sites</i> .
kpdu5_update_site_extensions	Tabela de mapeamento que liga <i>kpdu5_extensions</i> com <i>kpdu5_updates</i> .
kpdu5_usergroups	Armazena os dados referentes aos grupos de utilizadores definidos no <i>site</i> .
kpdu5_users	Armazena os dados referentes aos utilizadores definidos no <i>site</i> .
kpdu5_user_profiles	Contém uma linha para cada combinação de ID de utilizador e campo perfil. Esta tabela é usada se o <i>plugin Utilizador-Perfil</i> estiver activo.
kpdu5_user_usergroup_map	Guarda os dados de cada grupo que o utilizador é membro.
kpdu5_weblinks	Esta tabela guarda os dados de cada <i>weblink</i> definido no <i>site</i> .

### 2.2.3. Processo de actualização do CMS

Depois de se conhecer a estrutura interna e a interface administrativa do Joomla, pode-se avançar para o processo de actualização deste CMS.

Na versão 3 do Joomla, para actualizar é necessário ir do menu *Componentes » Actualização Joomla!* e clicar no botão *Instalar actualização*. É iniciado o processo de actualização que dura alguns minutos e ao terminar este processo deve-se limpar a cache através do botão *Purgar Cache*. [10]

### 2.2.4. Construção de *Plugins*

Segundo [11], um *plugin* é um programa de computador usado para adicionar alguma funcionalidade específica a um outro programa.

A maioria dos *plugins* consistem em apenas um único ficheiro de código, mas para uma instalação correcta do mesmo deve-se empacotá-lo num ficheiro de instalação que possa ser

processado pelo instalador Joomla. Seguem-se os passos necessários para desenvolver um *plugin* para o Joomla. [12]

## 1. Criar o ficheiro de instalação

No Joomla os *plugins* são facilmente instalados como ficheiros *.zip* (*.tar* e *.gz* também são suportados). Dentro do ficheiro *.zip* deve ser incluído um ficheiro de instalação XML correctamente formatado. Como exemplo, segue-se o código do ficheiro XML para um *plugin* da categoria de pesquisa. [12]

```
<?xml version="1.0" encoding="utf-8"?>
<extension version="3.1" type="plugin" group="search">
  <name>plg_search_categories</name>
  <author>Joomla! Project</author>
  <creationDate>November 2005</creationDate>
  <copyright>Copyright (C) 2005 - 2013 Open Source Matters. All rights reserved.</copyright>
  <license>GNU General Public License version 2 or later; see LICENSE.txt</license>
  <authorEmail>admin@joomla.org</authorEmail>
  <authorUrl>www.joomla.org</authorUrl>
  <version>3.1.0</version>
  <description>PLG_SEARCH_CATEGORIES_XML_DESCRIPTION</description>
  <files>
    <filename plugin="categories">categories.php</filename>
    <filename>index.html</filename>
  </files>
  <languages>
    <language tag="en-GB">en-GB.plg_search_categories.ini</language>
    <language tag="en-GB">en-GB.plg_search_categories.sys.ini</language>
  </languages>
  <config>
    <fields name="params">
      <fieldset name="basic">
        <field name="search_limit" type="text"
          default="50"
          description="JFIELD_PLG_SEARCH_SEARCHLIMIT_DESC"
          label="JFIELD_PLG_SEARCH_SEARCHLIMIT_LABEL"
          size="5"
        />
        <field name="search_content" type="radio"
          default="0"
          description="JFIELD_PLG_SEARCH_ALL_DESC"
          label="JFIELD_PLG_SEARCH_ALL_LABEL"
        >
          <option value="0">JOFF</option>
          <option value="1">JON</option>
        </field>
        <field name="search_archived" type="radio"
          default="0"
          description="JFIELD_PLG_SEARCH_ARCHIVED_DESC"
          label="JFIELD_PLG_SEARCH_ARCHIVED_LABEL"
        >
          <option value="0">JOFF</option>
          <option value="1">JON</option>
        </field>
      </fieldset>
    </fields>
  </config>
</extension>
```

Figura 3 - Código XML do ficheiro de instalação do *plugin* [12]

A entrada *group="search"* na tag *<extension>* e mais abaixo a tag *<filename>*, contêm a informação que diz ao Joomla em que pasta deve-se copiar o ficheiro e em que grupo deve ser adicionado o *plugin*. [12]

Se o *plugin* a ser criado responde a eventos existentes do núcleo, o atributo *group="xxx"* deve ser alterado para reflectir o nome da pasta do *plugin* existente para o tipo de evento que se deseja incrementar. Por exemplo, *group="authentication"* ou *group="user"*. Ao criar um novo *plugin* para responder a eventos do núcleo, é importante que o nome do *plugin* seja único para não entrar em conflito com qualquer um dos outros *plugins* que podem também estar respondendo ao evento núcleo que se deseja atender. [12]

**Nota:** Adicionar o atributo *method="upgrade"* para a *tag extension*, significa que o *plugin* pode ser instalado sem desinstalar uma versão anterior. Todos os ficheiros existentes serão substituídos, mas os ficheiros antigos não serão apagados. [12]

## 2. Criar o plugin

O modo orientado a objetos de escrever *plugins* envolve escrever uma subclasse de ***JPlugin*** – classe base que implementa as propriedades básicas dos *plugins*. Nos seus métodos, as seguintes propriedades estão disponíveis [12]:

- ✓ *\$this->params*: Os parâmetros estabelecidos para este *plugin* pelo administrador
- ✓ *\$this->\_name*: O nome do *plugin*
- ✓ *\$this->\_type*: O grupo (tipo) do *plugin*
- ✓ *\$this->db*: O objecto db (desde a versão Joomla 3.1)
- ✓ *\$this->app*: O objecto de aplicação (desde a versão Joomla 3.1)

No seguinte exemplo de código, **Figura 4**, **<PluginGroup>** representa o grupo (tipo) do *plugin*, e **<PluginName>** representa o seu nome. Note-se que os nomes das classes e funções em PHP diferenciam maiúsculas de minúsculas. [12]

E, assim, seguindo o tutorial acima descrito, é possível construir não apenas um *plugin* de pesquisa, como também outros tipos de *plugin*, tendo em conta a estrutura (*tags*) do ficheiro XML, e adaptando os dados dos parâmetros referentes ao *plugin* a desenvolver. Para o ficheiro de PHP, o mesmo não se aplica, visto que, este varia claramente consoante o tipo de *plugin*, pois a função utilizada num *plugin* de pesquisa, não será a mesma para um *plugin* de perfil de utilizador.

```

<?php
// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

class plg<PluginGroup><PluginName> extends JPlugin
{
    /**
     * Load the language file on instantiation. Note this is only available in Joomla 3.1 and higher.
     * If you want to support 3.0 series you must override the constructor
     */
    * @var    boolean
    * @since  3.1
    */
    protected $autoloadLanguage = true;

    /**
     * Plugin method with the same name as the event will be called automatically.
     */
    function <EventName>()
    {
        /*
         * Plugin code goes here.
         * You can access database and application objects and parameters via $this->db,
         * $this->app and $this->params respectively
         */
        return true;
    }
}
?>

```

Figura 4 - Código PHP da função do *plugin* [12]

### 2.3. DRUPAL

Desenvolvido por Dries Buytaert, o Drupal é uma plataforma de código aberto, que permite a construção de *sites* flexíveis e robustos. Uma das razões de ser designado como um dos melhores CMS é por permitir ao utilizador actualizar as suas páginas *web* sem conhecimentos técnicos e garantir que o *site* se encaixa no fluxo de trabalho da organização. No entanto, os programadores deste sistema alertam sobre a curva de aprendizagem, pois exige um esforço significativo compreendê-lo completamente devido à sua complexidade e dificuldade de personalização. [7] [8]

Trata-se de uma plataforma dinâmica que cresce à medida das necessidades/requisitos da organização ou utilizador individual. Tal como o Joomla, o Drupal também tem uma comunidade de programadores que oferecem suporte e documentação para o *website* a ser criado.

Algumas das suas principais características são [7] [13]:

- ✓ Construção de páginas;
- ✓ Criação de conteúdo e de utilizadores;
- ✓ *Design* e *Exibição*, com a nova API o utilizador controla exactamente o que é mostrado e em que posição do *site*.
- ✓ Expansão, mais de 800 módulos disponíveis para estender e adicionar funcionalidades.


Segue-se a apresentação da interface de administração com uma breve legenda elucidativa, e a estrutura interna do Drupal 7, nas secções 2.3.1 e 2.3.2, respectivamente.

### 2.3.1. Interface de administração

A **Figura 5** mostra a interface de administração do Drupal. Para auxiliar a apresentação da interface recorreu-se ao uso da numeração para os menus.



Figura 5 - Interface de administrador do Drupal

1.  – Direciona o administrador para o *front-end* do *site*.
2. **Painel de controlo** – Permite ver e personalizar o painel de controlo. É possível escolher que blocos se quer ver de modo a ter um rápido acesso. Por exemplo, ao escolher os blocos conteúdo recente, formulário de pesquisa e novos utilizadores, sempre que for ao painel de controlo, estes serão exibidos, bem como as informações neles presentes.
3. **Conteúdo** – Gestão e criação de conteúdos. Por defeito este menu permite criar artigos e páginas, através do *link* “Adicionar conteúdos”. Após a instalação de módulos de conteúdo e a activação dos mesmos, estes também aparecem como opção no *link* “Adicionar conteúdos”.
4. **Estrutura** – Permite administrar blocos, tipos de conteúdo, menus, etc.
5. **Aparência** – Permite instalar, seleccionar e configurar os *templates*.
6. **Pessoas** – Adicionar e gerir contas de utilizadores, grupos de utilizadores e as suas permissões.
7. **Módulos** – Permite ampliar as funcionalidades do *site*, por meio de instalações de módulos (*plugins*) como por exemplo, módulos de formulários, de idiomas, segurança, entre outros.
8. **Configuração** – Acesso às configurações das definições de administração.
9. **Relatórios** – Permite ver relatórios, actualizações e erros existentes.
10. **Ajuda** – Ajuda sobre a utilização e configuração de módulos (por exemplo: blocos, *dashboard*, etc.).
11. **Olá admin** – Dá acesso às definições da conta do administrador.
12. **Encerrar sessão** – Encerra a sessão de administrador.

### 2.3.2. Estrutura interna do Drupal

A API da base de dados do Drupal 7 [14] fornece uma camada de abstracção padrão, independente do fornecedor para aceder a servidores de base de dados. A API é projectada para preservar a sintaxe e o poder do SQL tanto quanto possível, mas também:

- Para suportar vários servidores de base de dados com facilidade;

- Para permitir aos programadores aproveitar funcionalidades mais complexas, como transacções;
- Para fornecer uma interface estruturada para a construção dinâmica de consultas;
- Para impor controlos de segurança e outras boas práticas;
- Para fornecer módulos com uma interface limpa para interceptar e modificar consultas de um *site*.

A principal documentação da API da base de dados [14] é derivada directamente de comentários no código.

A base de dados consiste em 73 tabelas e o seu diagrama encontra-se no **Anexo 1** uma vez que a imagem é demasiado grande.

### 2.3.3. Processo de actualização do CMS

Depois de se conhecer a estrutura interna e a interface administrativa do Drupal, pode-se avançar para o processo de actualização deste CMS.

Seguem-se as instruções sobre como actualizar o Drupal *core* com uma nova versão do Drupal da mesma categoria ou mesma versão base, isto é, Drupal 6, 7 e 8. [15]

1. Fazer uma cópia de segurança da pasta/ficheiros do *website* e da base de dados.
2. Fazer o download da versão de actualização e extrair o pacote (*unzip*).
3. Colocar o *site* em modo de manutenção.
4. Excluir todos os ficheiros e pastas dentro da instância original (pasta original do *website*) do Drupal, excepto a pasta *Sites* e quaisquer ficheiros personalizados que tenham sido adicionados noutros lugares.
5. Copiar todas as pastas e ficheiros, excepto a pasta *Sites* do pacote extraído, para dentro da instância original do Drupal.
6. Se a versão de actualização inclui alterações para o ficheiro *settings.php* deve-se substituir o ficheiro antigo no directório *.../sites/default/* pelo novo, e editar as entradas específicas do *website*, como o nome da base de dados, utilizador e *password*.
7. Se os ficheiros *.htaccess* ou *robots.txt* do directório original foram modificados, deve-se reaplicar essas alterações para os novos ficheiros.
8. Entrar no *site* como administrador e executar o ficheiro *update.php* através do URL *http://...nomeDoSeuWebsiteDrupal/update.php* (por exemplo: *http://localhost/site\_drupal723/update.php*).
9. Segue-se o processo de actualização.
10. Desactivar o modo de manutenção.

### 2.3.4. Construção de *Plugins*

No Drupal os módulos são *plugins* que estendem as funcionalidades deste CMS. Segue-se um tutorial de como criar um módulo personalizado em poucos passos. [16]

#### 1. Criação do directório do módulo

Ir ao directório dos módulos (*/sites/all/modules*) e criar uma pasta com o nome “*custom*”. Dentro da mesma criar um outro directório com o nome do módulo, por exemplo, “*person*”. Então o caminho do directório final será */sites/all/modules/custom/person*. [16]

#### 2. Ficheiros necessários para o módulo

É necessário criar pelo menos 2 ficheiros para a criação do módulo, um para as informações (com a extensão *.info*) e outro para o módulo (com a extensão *.module*). Então seguindo o exemplo, deve-se criar um ficheiro *person.info* e outro *person.module*. É de notar que o nome dos ficheiros deve coincidir com o nome do módulo. [16]

##### 2.1. Estrutura do ficheiro *.info*

```
1.      ; $Id $
2.      name = person
3.      description = about a person
4.      core = 6.x
```

Figura 6 - Estrutura do ficheiro *.info* [16]

##### 2.2. Campos adicionais do ficheiro *.info*

Suponha que o módulo precisa da versão 5 do php, então deve-se adicionar *php=5.0* ou que o módulo não será executado sem a habilitação do caminho do mesmo, então é necessário adicionar *dependencies[]=path*. Assim irá aparecer uma opção adicional “*Required by: person*” no módulo da listagem de página (módulo *Path*), como mostra a **Figura 7**. [16]



Figura 7 – Opção adicional no módulo *Path* [16]

Então o conteúdo do ficheiro *person.info* será

```
1.      ; $Id $
2.      name = person
3.      description = about a person
4.      core = 6.x
5.
6.      package = "custom module"
7.      dependencies[] = path
8.      php = 5.2
```

Figura 8 - Conteúdo do ficheiro *person.info* [16]

Para uma melhor compreensão da informação presente no ficheiro *person.info* ver a **Figura 9**.

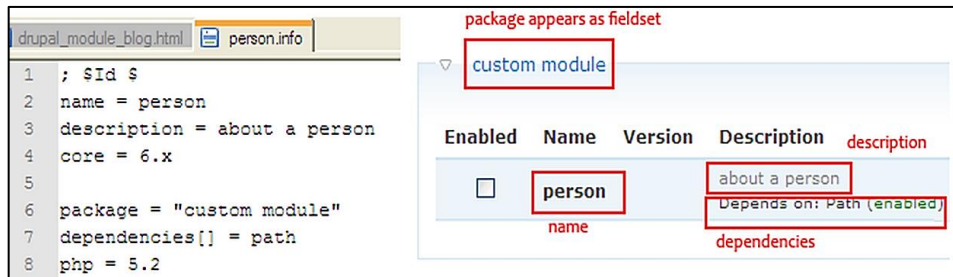


Figura 9 - Resultado do ficheiro *person.info* [16]

### 3. Criando o menu

Para criar um menu no *front-end* do *site* para o módulo é necessário usar a função do Drupal *hook\_menu()*, em que *hook* indica o nome do módulo, ou seja, a função será *person\_menu()*. Por fim, é essencial escrever o seguinte código dentro do ficheiro *person.module*: [16]

```

1. <? Php
2. // $ Id: person.module
3.
4. / **
5. * Implementos hook_menu ()
6. * /
7. função person_menu ( ) {
8.     $ items = gama ( ) ;
9.     $ items [ 'pessoa' ] = conjunto (
10.         'Title' => "Pessoa" ,
11.         'Page callback' => "person_personal_info" , // depois da visita
12.         'Page callback' => "person_personal_info function () é chamado
13.         'Callback acesso' => verdade , // deve retornar verdadeiro, c
14.         'Tipo' => MENU_NORMAL_ITEM, // drupal do tipo de menu padrão
15.         'Peso' => '10 ' , // que queremos mostrar pessoa link abaixo em
16.         nosso menu de navegação
17.     ) ;
18.     retornar $ items ; // finalmente, não se esqueça de retornar array $ i
19.     tens
20. }
21. >

```

Figura 10 - Código PHP do ficheiro *person.module* [16]

#### 3.1. Criando a função de retorno

Esta função irá gerar a informação a ser visualizada ao clicar no *link Person*. Como exemplo, a função terá algumas informações básicas sobre uma pessoa [16].

```

1. /**
2.  * callback function for person
3.  *
4.  */
5. function person_personal_info() {
6.     $output = 'Name: Hasan Hafiz </br>';
7.     $output .= 'City: Dhaka </br>';
8.     $output .= 'Country: Bangladesh </br>';
9.     return $output;
10. }

```

Figura 11 - Código da função de retorno [16]

#### 4. Activando o módulo

Ir ao menu *Module* e activar o módulo *Person*, de seguida ir para o *front-end* do *site* e clicar no *link Person* para visualizar a função de retorno [16].

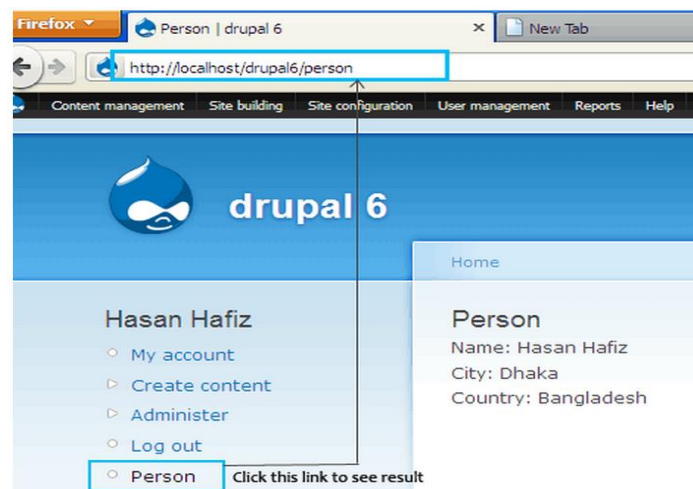


Figura 12 - Visualização da funcionalidade do *plugin* no *front-end* do *site* [16]

E assim, seguindo o tutorial acima descrito, é possível construir não apenas um *plugin* de perfil de utilizador, como também outros tipos de *plugin*, tendo em conta a estrutura (*tags*) do ficheiro *.info*, e adaptando os dados dos parâmetros referentes ao *plugin* a desenvolver. Para o ficheiro *.module*, o mesmo não se aplica, visto que, este varia claramente consoante o tipo de *plugin*, pois a função utilizada num *plugin* de perfil de utilizador, não será a mesma para um *plugin* de pesquisa, por exemplo.

#### 2.4. WORDPRESS

Originalmente construído como uma plataforma de *blogging*, desde 2009 tem vindo a mudar e a proporcionar uma boa gestão de conteúdo, tornando-se cada vez mais um CMS abrangente. Uma das suas vantagens é o grande número de *plugins* na sua comunidade,

fornecidos por programadores independentes. No WordPress os *plugins* são na verdade *add-ons* (complementos) que visam melhorar as funcionalidades da interface do utilizador, sendo esta a característica que o torna popular.

Algumas outras características do WordPress são [7] [17]:




- ✓ Total conformidade com padrões;
- ✓ Fácil importação de dados;
- ✓ Ferramentas de comunicação entre blogues, úteis para o utilizador se conectar a outros blogues e vice-versa;
- ✓ Protecção contra *spam*;
- ✓ Mensagens protegidas por *password*, para compartilhar artigos com algumas pessoas;
- ✓ Fluxo de trabalho.

### 2.4.1. Interface de administração

A **Figura 13** mostra a interface de administração do WordPress. Para auxiliar a apresentação da interface recorreu-se ao uso da numeração para os menus.



Figura 13 - Interface de administrador do WordPress

1.  – É o menu de ajuda. Fornece ligação ao *site* oficial do WordPress, documentação e fórum de suporte.
2. **Centro da Mãe** – Este menu tem o nome atribuído ao *site*, que neste caso é **Centro da Mãe** e direcciona o administrador para o *front-end* do mesmo.
3.  – Avisa se há actualizações e quantas existem. Direciona o administrador para a página das actualizações.
4.  ou **Comentários** – Direciona para a página dos comentários (feitos no *front-end* do *site*) que aguardam a aprovação do administrador.

5. Adicionar – Atalho para adicionar novos artigos, páginas, multimédia e utilizadores.
6. Olá, admin! – Permite editar as definições da conta do administrador e encerrar a sessão.
7. Painel – Acesso às configurações sobre o *site* e actualizações do WordPress.
8. Artigos – Permite criar e gerir artigos e as suas categorias e etiquetas.
9. Multimédia – Adicionar ficheiros de multimédia.
10. Páginas – Criar e gerir as páginas do *site*.
11. Apresentação – Personalizar a aparência do *site* (*templates*, *widgets*, cabeçalho, etc.)
12. *Plugins* – Instalar e gerir *plugins*.
13. Utilizadores – Adicionar e gerir contas de utilizadores.
14. Ferramentas – Permite importar e exportar conteúdo do/para o *site*. Por exemplo, artigos e comentários noutra sistema, o WordPress pode importá-los para o *site* WordPress.
15. Opções – Acesso às opções de configuração de administração.

#### 2.4.2. Estrutura interna do WordPress

Segue-se uma descrição resumida das tabelas criadas durante a instalação padrão do WordPress Versão 3.4 juntamente com o Diagrama Entidade-Relação (E-R) da base de dados com as relações entre as tabelas. Esta não é a versão usada para a avaliação dos CMSs, no entanto a informação não deixa de estar actualizada já que se verificou que na versão 3.6.1 a estrutura mantém-se.

Actualmente, a única base de dados suportada pelo WordPress é MySQL versão 5.0.15 ou superior. [18]

A base de dados consiste em 11 tabelas e o seu diagrama encontra-se no **Anexo 2** uma vez que a sua visualização ficaria um tanto pequena se inserida aqui.

Para uma melhor interpretação segue-se a **Tabela 2** com informações específicas do que está em cada tabela da base de dados do WordPress.

Tabela 2 - Descrição das tabelas do Diagrama E-R [18] [19]

WordPress 3.4 – 11 Tabelas	
Nome da tabela	Descrição
wp_commentmeta	Cada <b>comentário</b> apresenta informações adicionais ( <b>meta-dados</b> ) que são armazenadas na tabela <i>wp_commentmeta</i> .
wp_comments	Esta tabela armazena os <b>comentários</b> .
wp_links	Contém informações relacionadas com as <b>ligações</b> ( <i>links</i> ).
wp_options	Armazena as opções definidas no âmbito do painel <b>Administração &gt; Configurações</b> , tais como, nome, descrição, admin, <i>email</i> , o <i>login</i> do servidor de correio, etc.
wp_postmeta	Os <b>meta-dados</b> de cada <b>post</b> são armazenados em <i>wp_postmeta</i> . Alguns <i>plugins</i> podem adicionar as suas próprias informações a esta tabela.
wp_posts	O núcleo dos dados WordPress são as <b>mensagens/posts</b> . Estas são armazenadas na tabela <i>wp_posts</i> . Páginas e itens de menu de navegação também são armazenados nesta tabela.
wp_terms	As <b>categorias</b> para as mensagens e <i>links</i> , e as <b>etiquetas</b> para as mensagens são encontrados dentro da tabela <i>wp_terms</i> .
wp_term_relationships	<b>Mensagens</b> são associadas com as categorias ou <i>tags</i> da tabela <i>wp_terms</i> , e estas associações são mantidas na tabela <i>wp_term_relationships</i> . A associação de <b>ligações</b> para as suas respectivas categorias também são mantidas na tabela.
wp_term_taxonomy	Esta tabela descreve a taxonomia ( <b>categoria</b> , <i>links</i> , ou <b>tags</b> ) para as entradas na tabela <i>wp_terms</i> .
wp_usermeta	Cada <b>utilizador</b> possui informações chamado de <b>meta-dados</b> e são armazenadas em <i>wp_usermeta</i> .
wp_users	A lista de <b>utilizadores</b> é mantida na tabela <i>wp_users</i> .

### 2.4.3. Processo de actualização do CMS

Quando uma nova versão do WordPress está disponível o administrador recebe uma mensagem de actualização no seu ecrã de administrador do WordPress. Para proceder à actualização basta clicar no *link* dessa mensagem e o processo irá se completar automaticamente. Se por alguma razão não funcionar ou se preferir, pode-se fazer a actualização pelo processo manual que é um pouco mais trabalhoso.

O método manual consiste em alguns passos [20]:

1. Substituir os ficheiros do WordPress.
  - 1.1. Descarregar a última versão do WordPress.
  - 1.2. Extrair o ficheiro .zip que descarregou.
  - 1.3. Desactivar os *plugins*.
  - 1.4. Excluir os antigos directórios *wp-includes* e *wp-admin* presentes na pasta do seu *website*.

- 1.5. Colocar os directórios *wp-includes* e *wp-admin* do ficheiro descarregado na pasta do *website*.
  - 1.6. Enviar os ficheiros individuais da nova pasta *wp-content* para a pasta *wp-content* existente, substituindo os ficheiros existentes. Não excluir a pasta *wp-content* existente no directório do *website*, nem eliminar quaisquer ficheiros ou pastas no directório *wp-content* existente (excepto os ficheiros que estão a ser substituídos).
  - 1.7. Carregar todos os novos ficheiros soltos a partir do directório raiz da nova versão para o directório raiz do WordPress existente.
  - 1.8. Apagar o ficheiro *.maintenance* (se estiver a actualizar manualmente após uma auto-actualização com falha, excluir o ficheiro *.maintenance* do directório WordPress utilizando o FTP. Isto irá remover a mensagem “actualização falhou”).
2. Actualizar a instalação.
  3. Visitar a página de administração do WordPress em */wp-admin*. Pode ser necessário iniciar sessão novamente. Se uma actualização de base de dados for necessária, neste ponto, o WordPress irá detectá-lo e gerar um URL, como por exemplo, <http://example.com/wordpress/wp-admin/upgrade.php>. Seguindo o *link* e as instruções, a base de dados será actualizada para ser compatível com a última versão do código. É importante fazer isso o mais rápido possível após o passo 1).
  4. Após a instalação deve-se limpar a *cache*.
  5. Por último deve-se voltar a activar os *plugins* e limpar os cookies do *browser*.

É de salientar que deve sempre fazer-se uma cópia de segurança do directório do *site* e da sua base de dados, tanto em actualizações automáticas como manuais. Por defeito o WordPress disponibiliza uma notificação ao administrador com a seguinte mensagem “**Importante:** antes de actualizar, por favor faça uma cópia de segurança dos seus ficheiros e base de dados. Para mais ajuda com as actualizações, visite a página do *Codex* sobre actualização WordPress.”

## 2.5. COMPARAÇÃO DOS CMS

Os dados que se seguem encontram-se em [7].

### 2.5.1. O mais pesquisado

A **Figura 14** representa a popularidade dos CMS em pesquisas realizadas no *Google*, entre os utilizadores, bem como os CMS que fornecem mais documentação, visto que uma melhor documentação consegue mais utilizadores e boas comunidades.

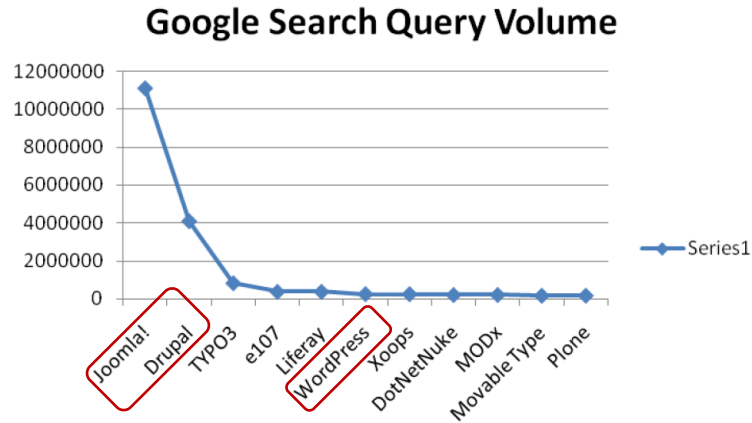


Figura 14 - Número de Pesquisas no Google [7]

De entre os 3 sistemas aqui estudados é visível que o mais popular é o Joomla, seguindo-se o Drupal e o WordPress. No entanto, isto não é suficiente para se poder afirmar qual o melhor CMS de todos. Existem outros aspectos a considerar. [7]

### 2.5.2. Complexidade

O Drupal é o CMS mais complexo, quanto à facilidade de utilização para um utilizador sem experiência ou sem conhecimentos em programação, em comparação com o Joomla e o WordPress. O Joomla é um sistema simples que não exige uma instalação complexa e é muito fácil de usar. É ideal para pequenas empresas que não requerem complexidade. No entanto, há menor possibilidade de estender o *site*.

### 2.5.3. Recursos

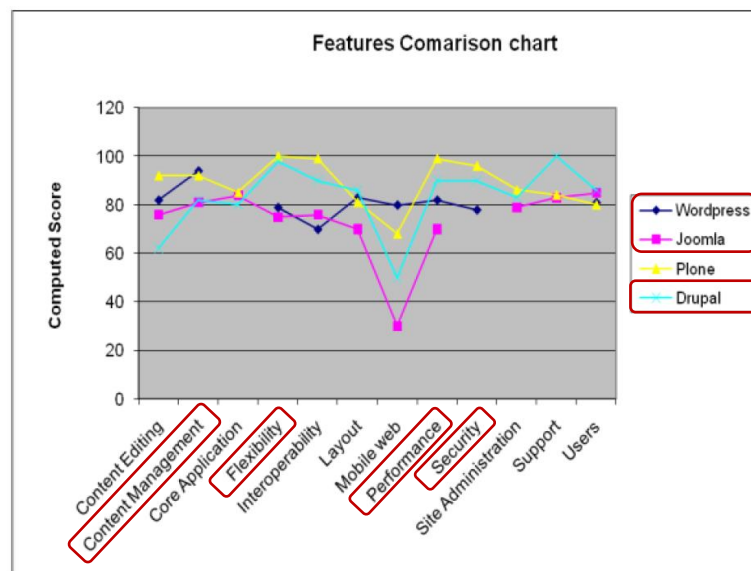


Figura 15 - Comparação de recursos/características [7]

No que diz respeito a recursos de gestão de conteúdo, os CMS Joomla e Drupal estão na mesma classificação. Mas olhando para a flexibilidade vemos que os menos flexíveis são o Joomla e de seguida o WordPress e o Drupal. O mesmo acontece no desempenho. No que toca à segurança, Joomla nem aparece no gráfico, sendo este então considerado o menos ou nada seguro. Depois surge o WordPress, e como mais seguro temos então o Drupal. [7]

#### 2.5.4. Popularidade

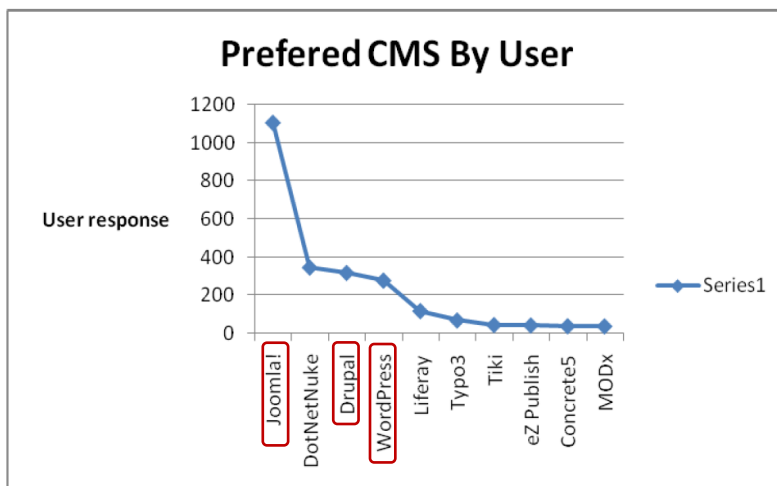


Figura 16 - CMS preferido dos utilizadores [7]

Segundo uma pesquisa sobre qual o CMS preferido entre os utilizadores podemos ver na **Figura 16**, que o Joomla foi o grande vencedor ocupando a 1ª posição, Drupal ficou na 3ª posição e o WordPress na 4ª posição, estes dois últimos com uma diferença considerável em relação ao Joomla. [7]

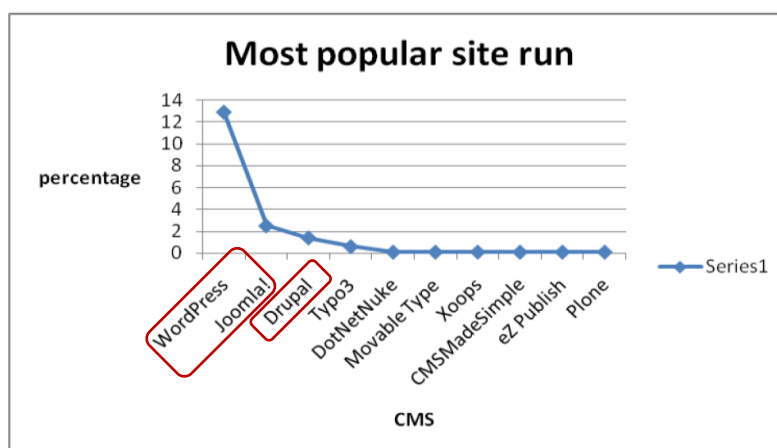


Figura 17 - O CMS mais utilizado [7]

De acordo com o *ranking* Alexa, – conhecido por fornecer informação sobre *websites*, incluindo o top dos *sites*, estatísticas sobre o tráfego de *internet*, etc. – os *sites*

(desenvolvidos usando um CMS) mais populares em execução são os *sites* do WordPress com mais de 12.5%, em 2º lugar com quase 2.5% está o Joomla e em 3º lugar o Drupal com 1.4%. [7]

## 2.5.5. Vantagens e Desvantagens

### 2.5.5.1. Joomla

#### Vantagens

- ✓ Amigável ao utilizador: Não tão amigável como o WordPress, mas ainda é relativamente fácil de usar;
- ✓ Variabilidade de extensão: No Joomla as extensões são divididas em 5 categorias: componentes, *plugins*, *templates*, módulos e idiomas;
- ✓ Interface: Tem uma interface intuitiva.

#### Desvantagens

- ✓ Curva de aprendizagem: A curva de aprendizagem não é íngreme, mas o utilizador tem de estar familiarizado com o sistema, o que implica alguma aprendizagem;
- ✓ Falta de *plugins* gratuitos: Não existem muitos *plugins* disponíveis gratuitamente.

Uso Recomendado: Para PME de comércio electrónico, permite criar um *site* com mais estabilidade estrutural. [21]

Pela descrição do *site* oficial do Joomla [22] e alguns artigos relacionados [7] [8] [21], este parece ser muito fácil de usar, o que desperta uma certa curiosidade até mesmo a um utilizador comum, ou seja, com apenas alguns conhecimentos básicos na área da informática.

Embora o *site* oficial do Joomla disponibilize documentação, esta não é muito útil pois não contém exemplos práticos, e não há nada como a prática para se aprender. Sendo assim, para se poder criar um *site* com o Joomla, é necessário algum tempo de dedicação a explorar o sistema. Também existem vídeo-aulas na *internet*, por exemplo [23] [24], publicadas por utilizadores independentes que ajudam muito mais do que a documentação disponibilizada pelo próprio *site* Joomla.

### 2.5.5.2. Drupal

#### Vantagens

- ✓ Extremamente flexível: É capaz de lidar tanto com um *site* simples com páginas estáticas, como com um poderoso *back-end* que suporta centenas de milhares de páginas e milhões de utilizadores;

- ✓ **Estabilidade:** Estável mesmo quando serve milhares de utilizadores simultaneamente.

#### Desvantagens

- ✓ **Usabilidade:** Não muito amigável a utilizadores novatos;
- ✓ **Curva de aprendizagem:** Extremamente difícil de usar para um uso frequente. O utilizador tem de ter capacidades de programação e ler imensos documentos técnicos;
- ✓ **Falta de *plugins* gratuitos:** No Drupal os *plugins* são chamados de módulos e a maioria deles não são gratuitos;
- ✓ **Falta de temas:** Não existe muita variedade de *templates*. Se o utilizador quer dar outro aspecto visual ao site terá de criar um *template* de raiz.

Uso recomendado: Para grandes projectos onde a estabilidade, escalabilidade e poder são prioritários sobre a facilidade de uso e estética. [21]

Numa publicação de Julho de 2013 [21], o critério “Amigável ao utilizador” foi referenciado como uma vantagem, no entanto pela experiência pessoal do autor desta tese e por alguma pesquisa, isso não se verifica, uma vez que criar um *site* com o Drupal requer alguns conhecimentos mais avançados e exploração do sistema, que só um utilizador técnico poderá ter. Além disso, a justificação do critério “Curva de aprendizagem”, que surge nessa mesma publicação, contradiz um pouco o critério “Amigável ao utilizador” referido como vantagem pela autora.

Outro aspecto a salientar é a formatação do texto na introdução de conteúdo. O Drupal permite a introdução de *tags* HTML, no entanto ao visualizar o resultado final no *site* verificou-se que estas nem sempre originam o efeito desejado. A solução para esta lacuna é a instalação de alguns módulos de formatação de texto – módulos *WYSIWYG* –, disponíveis gratuitamente no *site* oficial do Drupal. Contudo, à data da escrita deste documento, estes não funcionavam para a versão 7 do Drupal, pois ainda não tinham sido actualizados como se pode verificar em [25].

#### **2.5.5.3. WordPress**

A seguinte informação foi retirada de [21], no entanto é de clarificar uma “contradição” em relação ao que se diz nas vantagens e no uso recomendado. Nas vantagens para o WordPress é dito que com este CMS pode ser feito praticamente qualquer coisa, ou seja, vários tipos de *site* (além de blogues), mas de acordo com o que é apresentado no uso recomendado é preciso ter em atenção que se for um *site* complexo o WordPress pode não funcionar muito bem.

### Vantagens

- ✓ Enorme biblioteca de *plugins*: O WordPress desenvolveu centenas de milhares de *plugins* gratuitos;
- ✓ Amigável ao utilizador: A interface é fácil de usar e altamente intuitiva;
- ✓ Flexibilidade: Graças aos *plugins* e temas personalizados, o WordPress torna-se apto à criação de vários tipos de *site*.

### Desvantagens

- ✓ Segurança: Frequentemente alvo de *hackers*. É necessária a instalação de *plugins* de terceiros para aumentar a segurança;
- ✓ Incompatibilidade com *plugins* mais antigos: Actualizações são muitas vezes incompatíveis com *plugins* mais antigos;
- ✓ Recursos de gestão de conteúdos limitados: Isto afecta a capacidade de lidar com grandes quantidades de conteúdo.

Uso Recomendado: Para soluções *blogging* simples ou outro tipo de *site* com nível de complexidade média. [21]

Apesar da enorme biblioteca de *plugins* disponível, estes por vezes têm problemas de compatibilidade. Ainda assim, no que diz respeito a funções de personalização, o WordPress disponibiliza poucas funções para que se possa fazer algo que não se pareça com um blogue. A falta de opções de estrutura e formatação, obriga a recorrer à linguagem HTML uma vez que este permite a sua introdução.

Outro problema a destacar, é impossibilitar o utilizador de comentar nas páginas do *site*. Uma possível solução seria desactivar essa opção através do menu Discussão, contudo por não surtir efeito foi necessário ir ao ficheiro *comments.php* comentar as linhas de código pertencentes à visualização do formulário de comentário no *site*.

Tendo-se apresentado na secção 2.5 a **Comparação dos CMS**, desde o mais pesquisado, à complexidade, recursos, popularidade e vantagens e desvantagens de cada um dos sistemas, é apresentado na **Tabela 3** uma breve comparação entre o Joomla, Drupal e WordPress.

Tabela 3 - Breve Comparação entre o Joomla, Drupal e WordPress

	Joomla	Drupal	WordPress
<b>Tipos de sites</b>	Pouco complexos	Complexos	Simple
<b>Curva de aprendizagem</b>	Média	Difícil	Fácil
<b>Flexibilidade</b>	Média	Alta	Limitada
<b>Escalabilidade</b>	Média	Alta	Limitada
<b>Interface utilizador</b>	Amigável	Pouco Amigável	Amigável

Na **Tabela 3** é possível visualizar para que tipo de *sites* cada CMS é aconselhado, a curva de aprendizagem de cada um deles, a flexibilidade, escalabilidade e o quão amigável é a interface de utilizador destes sistemas.

### 2.5.6. Ficha Técnica

Para uma visualização mais rápida e geral sobre cada um dos CMS a **Tabela 4** apresenta a ficha técnica dos mesmos, segundo [6].

Tabela 4 - Ficha técnica dos CMS líderes [6]

	Joomla	Drupal	WordPress
<b>Data de lançamento</b>	16/9/2005	15/1/2001	27/5/2003
<b>Nº de versões núcleo</b>	6	7	3
<b>Nº total de updates</b>	A partir da versão 1.5 e 1.6: <b>27</b> (incluindo beta)	<b>77</b>	<b>164</b>
<b>Frequência de actualizações de versão (em média)</b>	Versão 1.5: uma vez a cada 49 dias Versão 1.6: uma vez a cada 25 dias	Desde que a versão 4.3 foi lançada em Novembro de 2003: uma vez a cada 36 dias	Uma vez a cada 17,8 dias
<b>Nº de plugins/módulos/extensões</b>	7 609	8 039	14 629
<b>Nº de temas</b>	Desconhecido	885	1 392
<b>Visitas únicas mensais nos EUA para o site principal</b>	59 600	55 700	50 Milhões
<b>Nº de Websites usando a plataforma (dos principais milhões de sites)</b>	2.7%	1.6%	14.3%
<b>Sites populares que usam a plataforma</b>	1 – IHOP 2 – Linux.com 3 – Porsche	1 – The White house 2 – MTV 3 – London.gov.uk	1 – MTV Newsroom 2 – Time 3 – The Ford Story
<b>Custo de manutenção mensal média</b>	\$500	\$1500	\$250
<b>Pesquisas mensais globais do Google</b>	11.1 Milhões	5 Milhões	30.4 Milhões

### 2.6. ESCOLHA DOS CMS

Como mencionado na secção **1.2 Objectivos**, foi feita uma análise de cada CMS, identificação de características e comparação dos CMS. Durante este estudo teórico – feito através de pesquisa e consulta de artigos publicados – foi necessário conciliar um estudo prático, para uma melhor percepção e interacção com os sistemas. Este estudo baseou-se em explorar os CMS, executando as mesmas tarefas – como por exemplo, criação de

utilizadores, páginas e artigos –, descritas na secção **2.6.2**, em todos de forma a compará-los quanto à facilidade de utilização como utilizador e como administrador e quanto ao nível de cumprimento dessas tarefas.

Seguem-se as secções **2.6.1** e **2.6.2**, que descrevem as versões dos CMS utilizados e as tarefas realizadas para estudo dos mesmos, respectivamente.

### **2.6.1. Versões dos CMS**

Esta secção é dedicada à descrição das versões dos CMS usados para o estudo e desenvolvimento dos *sites*.

#### **2.6.1.1. Joomla 3.x**

O Joomla série 3 foi lançado em Setembro de 2012, com novos recursos como, a adaptação a dispositivos móveis, a instalação de pacotes de idiomas directamente do gestor de extensão, novos *templates*, salvar artigos em branco, melhorias para a pesquisa inteligente, entre outros. [26]

Passados sete meses é lançada a versão 3.1 do Joomla – versão utilizada neste projecto – que veio corrigir uma grande quantidade de erros, com um total de 242 correcções, nomeadamente, correcções na interface de administração e melhorias na qualidade do código. [27]

#### **2.6.1.2. Drupal 7**

Lançado a 5 de Janeiro de 2011 [28], esta versão do Drupal tem como requisitos mínimos:

1. Base de dados: MySQL 5.0.15 ou PostgreSQL 8.3
2. Versão PHP: 5.2 ou superior
3. Memória PHP: 40M - 64M

Esta versão *core* do Drupal usufrui de melhoramentos no sistema de segurança, bem como de usabilidade, como por exemplo, (1) configuração de execução de tarefas automáticas – avisos de actualizações de módulos; (2) redesenho da validação da força da *password* de modo a se tornar mais agradável; (3) aperfeiçoamento do suporte de fuso horário. No que diz respeito à base de dados da versão 7 do Drupal, foi introduzido o padrão para o motor InnoDB, em vez do MyISAM, no MySQL quando disponível para uma maior escalabilidade e integridade dos dados.

Outra novidade nesta versão é a manipulação de ficheiros e imagens, em que foi adicionado um campo específico, respectivamente, para o *upload* dos mesmos, deixando de ser necessária a instalação de módulos para esta tarefa.

[29]

### 2.6.1.3. WordPress 3.6

Lançado ao público a 1 de Agosto de 2013 [30], esta versão do WordPress vem com uma interface nova e mais simples na criação e edição de menus para o *site* e, também com um novo *template* denominado por “*Twenty Thirteen*”, que ao contrário dos seus antecessores, é um pouco mais ousado e colorido. O WordPress 3.6 disponibiliza uma funcionalidade de gravação automática, deixando o utilizador despreocupado no que diz respeito à perda de conteúdo que não tenha sido previamente gravado. [31]

A 11 de Setembro de 2013 o WordPress lança a versão 3.6.1 – versão utilizada no decorrer deste projecto –, que é uma versão de manutenção e actualização de segurança. Esta versão vem com robustez de segurança adicional, como por exemplo, a actualização das restrições de segurança em todo o carregamento de ficheiros, para minimizar o potencial de *cross-site scripting*<sup>3</sup>; as extensões *.swf* e *.exe* não são mais permitidas por padrão; e, as extensões *.htm* e *.html* só são permitidas se o utilizador tiver a capacidade de usar HTML não filtrado. [32]

## 2.6.2. Realização de tarefas

Com base na lista de requisitos (secção 3.1), foi possível verificar a necessidade de explicar determinadas tarefas uma vez que estas são recorrentes e relevantes para a gestão e manutenção do *site*.

Da secção 2.6.2.1 à 2.6.2.4.3 são explicadas algumas dessas tarefas para cada CMS.

### 2.6.2.1. Criar um utilizador

**2.6.2.1.1. Joomla** – Neste CMS a criação de utilizadores é feita através do menu *Utilizadores* » *Membros* » *Novo Utilizador*. Preenche-se o formulário sobre os detalhes de conta e no separador “*Grupos de utilizador*” atribuem-se as permissões de acordo com o tipo de utilizador associado. Os utilizadores podem pertencer ao grupo *front-end* (só têm acesso à área o *site*) ou ao grupo *back-end* (tem acesso ao *site* e à área de administração). No grupo *front-end* existem os seguintes tipos de utilizador: [33] [34]

- ✓ **Public** só pode ver informações que sejam do tipo *public* e ainda pode registar-se. No caso que não o tenha feito, este não tem acesso às áreas que são para utilizadores registados.

---

<sup>3</sup> **Cross-site scripting (XSS)** – tipo de vulnerabilidade do sistema de segurança de um computador, encontrado normalmente em aplicações *web* que activam ataques maliciosos ao injectarem *scripts* HTML do lado do cliente, dentro das páginas *web* vistas por outros utilizadores. [58]

- ✓ Guest tem acesso não só à informação do tipo *public* como também do tipo *registered*, restringindo-o da informação *special*, ou seja, não pode adicionar nem editar conteúdo.
- ✓ Author tem acesso à informação *public*, *registered* e *special*. Pode criar conteúdo, em que este está sujeito à aprovação e publicação por alguém da área de administração (Administrador ou Super Administrador), e também editar conteúdo mas que tenha sido criado por si próprio, isto é, não pode editar conteúdo criado por outro utilizador.
- ✓ Editor é uma derivação do *Author*, mas que pode editar todos os conteúdos.
- ✓ Publisher é um *Editor* mas com a possibilidade de fazer publicações de conteúdo.

No grupo *back-end* existem os seguintes tipos de utilizador:

- ✓ Manager pode criar, editar e eliminar conteúdo e categorias, mas apenas tem acesso à gestão de conteúdo e multimédia.
- ✓ Administrator é possível criar vários administradores. Estes têm acesso a tudo excepto à instalação de *templates* e ficheiros de linguagem, poder eliminar ou configurar a conta do Super Administrador, ter acesso à configuração global do *site*, às informações do sistema nem ao componente Email em massa.
- ✓ Super Administrator/Super Users só existe um e tem acesso a tudo sem restrições.

**2.6.2.1.2. Drupal** – No Drupal o menu de criação de utilizadores denomina-se por “Pessoas”. Para proceder à criação de utilizadores deve-se ir ao menu *Pessoas » Adicionar utilizador*, preencher o formulário com os detalhes do utilizador e atribuir as permissões no separador “Permissões”. Neste CMS as permissões também são atribuídas por grupos de utilizador – utilizador anónimo, utilizador autenticado e administrador. No entanto, este não “fixa” permissões por grupo, podendo assim o administrador alterar as permissões de cada grupo. Ou seja, qualquer tipo de utilizador pode ter quaisquer permissões, como por exemplo de administrador. Ainda assim, apesar do sistema não ter nenhum método que impeça tais atribuições de permissões, este mostra uma nota com o intuito de alertar que algumas permissões só devem ser atribuídas a utilizadores de confiança.

**2.6.2.1.3. WordPress** – A criação de um novo utilizador é feita no menu *Utilizador » Adicionar novo*, preenchendo o formulário com os dados do utilizador e atribuindo as permissões na secção “Perfil”. Os tipos de utilizador no WordPress são: [35]

- ✓ Administrador que tem acesso a todas as funcionalidades, sem restrições.

- ✓ Editor tem acesso à publicação e gestão de artigos e páginas, como também consegue gerir conteúdos publicados por outros utilizadores.
- ✓ Autor só pode gerir e publicar os seus próprios artigos.
- ✓ Colaborador pode criar e gerir os seus artigos mas não os pode publicar; estes têm de ser autorizados pelo “Editor” ou “Administrador”.
- ✓ Assinante pode visualizar conteúdo e apenas editar e visualizar o seu perfil de utilizador.

### 2.6.2.2. Criar uma página

**2.6.2.2.1. Joomla** – Para criar uma página é preciso ir ao menu *Menus » Menus* e é-lhe apresentada a lista de menus padrão (*main menu* e *user menu*) do Joomla. Deve-se escolher o “*main menu*” e clicar no botão “+Novo” para iniciar a criação da página. De seguida preenche-se o formulário com os dados da página a criar e no separador “Atribuição de módulos para este item de menu” é possível escolher que módulos vão estar presentes na página (por exemplo: menu *login*).

**2.6.2.2.2. Drupal** - Através do menu *Conteúdo » Adicionar novo conteúdo » Basic Page* é possível criar uma página no Drupal. Preenche-se o formulário com o nome da página no campo “*Title*” e o conteúdo da página no campo “*Body*”, que poderá conter *tags* HTML. Para a página ser também um menu do *site*, deve habilitar o campo “Fornecer uma ligação do menu” e em “Opções de publicação”, habilitar o campo “publicado” para que esta página fique visível no *front-end* do *site*.

**2.6.2.2.3. WordPress** - Através do menu *Páginas » Nova página* é possível criar uma nova página, e tal como no Drupal é necessário atribuir um título e adicionar o conteúdo que poderá ser em linguagem HTML, desde que seleccionado o separador “HTML”. A secção “Atributos da página” não é mais do que a escolha da posição que essa página irá ocupar na barra de menus do *site*.

### 2.6.2.3. Criar artigos

**2.6.2.3.1. Joomla** – A criação de artigos pode ser feita pelo menu *Conteúdos » Artigos » Criar artigo* onde é apresentado um formulário para introduzir o título do artigo e o seu conteúdo. O Joomla já fornece alguns elementos de formatação de texto de modo a facilitar e de certa forma evitando a utilização de linguagem HTML. No entanto, se o pretendido for criar um artigo com formatação mais complexa e com outros elementos (tabelas, etc.) para além do que o Joomla fornece, pode-se fazê-lo em HTML através do elemento “HTML” presente na barra de formatação.

**2.6.2.3.2. Drupal** – Neste CMS a criação de artigos é feita pelo menu *Conteúdo » Adicionar novo conteúdo » Article*. O cenário é o mesmo de quando se cria uma nova página mas com um novo campo “tags” que serve para denominar o artigo em forma de tag, sendo depois mais fácil de localizá-lo entre os outros artigos.

**2.6.2.3.3. WordPress** – Criar um artigo no WordPress é bastante intuitivo, é só clicar no menu *Artigos » Novo artigo*. O cenário é idêntico ao da criação de páginas mas com uma secção que permite escolher o formato do artigo, isto é, se o artigo terá o formato de citação, nota, link, entre outros. É de salientar que nem todos os templates suportam essa opção dos formatos de artigo. Templates de versões anteriores à 3.6.1 muito provavelmente não serão compatíveis.

### 2.6.2.4. Instalar Plugins

**2.6.2.4.1. Joomla** – A instalação de plugins é através do menu *Instalações » Gestor de instalações » Instalar* e consoante o tipo de ficheiro do plugin deve-se escolher o procedimento para a instalação, se é por meio de um pacote, se a partir de uma pasta ou por um endereço URL. Após a introdução do caminho do ficheiro, basta clicar no botão “Instalar” para dar início à instalação.

**2.6.2.4.2. Drupal** – No Drupal os plugins são denominados por módulos. Então para instalar um módulo é necessário ir ao menu *Módulos » Instalar novo módulo* e introduzir o endereço URL ou carregar o ficheiro – no caso de já o ter no computador – e por fim clicar no botão “Instalar”. Após a instalação é necessário proceder à activação do módulo.

**2.6.2.4.3. WordPress** – O menu *Plugins » Adicionar novo* permite a instalação de novos plugins, onde é necessário escolher o link “carregar” e seleccionar o ficheiro – que já deve estar descarregado no computador – e proceder à instalação.

### 2.6.2.5. Criar banner

- 2.6.2.5.1. Joomla** – Para inserir um *banner* é necessário ir ao menu *Componentes » Faixas Publicitárias » Novo* e preencher os respectivos campos do formulário. É de notar que para um *banner* estático deve-se introduzir o endereço URL da imagem no campo “endereço de destino”, caso seja um *banner* dinâmico – com mais de uma imagem – deve-se introduzir o código HTML no campo “Código personalizado”.
- 2.6.2.5.2. Drupal** – Para inserir um *banner* no Drupal é essencial criar um bloco para esse fim, o que implica ir ao menu *Estrutura » Blocos » Adicionar bloco*, preencher o formulário com a informação necessária e no campo “corpo do bloco” introduzir o código HTML – tanto para um *banner* dinâmico como para um estático. Depois do bloco criado, deve-se habilitá-lo na posição do *site* onde se deseja mostrar o *banner*.
- 2.6.2.5.3. WordPress** – Inserir um *banner* no WordPress pode ser pela criação de uma página ou artigo, em que no separador “HTML” se introduz o código HTML – que chama a imagem estática ou dinâmica – no conteúdo da página/artigo, ou então pelo menu *Apresentação » Widgets* e arrastar o elemento “Texto” para a secção “Área primária de *widgets*” – onde o *banner* aparecerá no rodapé do *site* – ou para a secção “Área secundária de *widgets*” – onde o *banner* surge em artigos e páginas na barra lateral direita.

Após o estudo, análise e comparação dos três CMSs é o momento de escolher dois deles e proceder à implementação do *site* para o Centro da Mãe.

Tendo em conta a existência de três critérios que um utilizador deve considerar na escolha de um CMS – (1) Exigência do *site*; (2) Necessidades futuras; e (3) Conhecimento técnico do utilizador –, os CMS escolhidos foram primeiramente o Joomla, pela sua interface fácil, clara e intuitiva, como também pela oferta de funcionalidades adicionais (*add-ons*), e entre o Drupal e o WordPress, este último foi excluído porque por mais *plugins* que possua a sua disposição prende-se sempre à estrutura de um blogue – i.e., publicação acrescida de comentário –, ficando assim o Drupal que apesar da sua interface de administração não ser tão amigável e intuitiva, respondia melhor às necessidades do *site* a desenvolver.



### 3. DESENVOLVIMENTO

Tendo como um dos objectivos desenvolver um *website* para uma IPSS, esta secção contém todo o processo da implementação do *website* para o Centro da Mãe – Associação de Solidariedade Social para apoiar grávidas adolescentes, jovens mães e seus filhos – em dois dos CMS estudados neste projecto, de forma a analisá-los e compará-los.

O *website* tem o intuito de divulgar a informação referente ao Centro da Mãe, de modo a abranger um maior número de pessoas, e ainda disponibilizar uma loja *online* para a instituição.

#### 3.1. LISTA DE REQUISITOS

O primeiro passo no desenvolvimento de um projecto é o levantamento de requisitos. Esta análise é o processo de descoberta, refinamento, modelagem e especificação do que é realmente importante para o desenvolvimento do sistema.

Após a primeira reunião com os *stakeholders*<sup>4</sup> foi possível obter a informação relevante para o problema e assim entender quais as reais necessidades do sistema.

As secções **3.1.1** e **3.1.2** apresentam a lista de requisitos funcionais e não-funcionais do sistema a ser desenvolvido.

##### 3.1.1. Requisitos Funcionais

###### 3.1.1.1. Do lado do sistema

1. Permite a introdução, alteração e eliminação de conteúdo.
2. Permite a introdução, alteração e eliminação de páginas.
3. Permite a introdução e remoção de utilizadores.
4. Permite a atribuição e eliminação de permissões de utilizadores.
5. Permite a inscrição de sócios.
6. Permite a inscrição de voluntários.
7. O utilizador tem de estar autenticado.
8. O utilizador administrador tem todas as permissões.
9. O utilizador que irá gerir o *site* poderá:
  - a. Ver conteúdo criado por qualquer utilizador;
  - b. Criar conteúdo;
  - c. Editar conteúdo criado por si ou por outro utilizador;

---

<sup>4</sup> **Stakeholders** – termo usado na área de informática referente à parte interessada do sistema, ou seja, os clientes.

- d. Eliminar conteúdo criado por si;
- e. Alterar imagens (por exemplo, de *banner* ou *slide* de imagens);
- f. Utilizar linguagem HTML no conteúdo.

#### 3.1.1.2. Do lado do site

- 10. O utilizador não tem de estar autenticado para aceder ao *site*.
- 11. O utilizador tem de estar autenticado para realizar compras na loja *online*.
- 12. O utilizador pode visualizar o conteúdo.
- 13. O utilizador pode inscrever-se como sócio.
- 14. O utilizador pode inscrever-se como voluntário.

#### 3.1.2. Requisitos Não-Funcionais

##### Acessibilidade

- 1. Conteúdo inserido ou alterado será actualizado em tempo real na BD.
- 2. É necessário ter ligação à internet para a visualização do *site*.

##### Desempenho

- 3. O tempo de resposta depende do *hardware* do utilizador.
- 4. O tempo de resposta depende da velocidade da *internet*.

##### Facilidade de uso

- 5. A interface do *site* deve ser simples, fácil de usar e intuitiva.

##### Portabilidade

- 6. O *site* deverá funcionar em qualquer *browser*.

### 3.2. SOLUÇÕES PROPOSTAS

De acordo com a lista de requisitos, descrita na secção 3.1 e os CMSs Joomla e Drupal, escolhidos para desenvolver o website, é importante referir que as funcionalidades padrão destes CMSs conseguem dar resposta a grande parte dos itens da lista de requisitos, sendo a realização das tarefas de implementação dos formulários e da loja virtual, dependentes de funcionalidades extra (plugins, componentes, módulos) disponibilizadas pelas respectivas comunidades.

No decorrer da implementação do *site* houve a necessidade de optar por algumas soluções para a realização de tarefas tais como, criação de formulários e da loja virtual, pois existem várias funcionalidades (*plugins*) disponíveis para o mesmo fim.

Para cada tarefa foram escolhidas duas soluções. Estas foram analisadas e estudadas de modo a escolher a que melhor se adequasse às necessidades do *site* e do utilizador.

Nas secções **3.2.1**, **3.2.2** e **3.2.3** são analisadas as soluções propostas para cada tarefa em cada CMS.

### **3.2.1. Formulários**

Tendo em conta os requisitos funcionais **5** e **6**, a criação de formulários é fundamental uma vez que as inscrições de sócio e voluntariado são feitas *online*.

Para a construção dos mesmos, no Joomla foram escolhidas as soluções *ChronoForms* e *FormMaker*, e no Drupal, *WebForm* e *FormMaker*.

#### **3.2.1.1. Joomla**

##### **A. ChronoForms**

*ChronoForms* é um componente Joomla que ajuda os utilizadores deste CMS a publicar formulários para um *site*. [36] [37]

Alguns dos poderosos recursos de personalização deste componente são: [36] [37]

1. Capacidade de usar *JavaScript* e *CSS* no código do formulário;
2. Utilização de código *PHP* no corpo do formulário ou quando submetido a fazer alguma acção do lado do servidor, como guardar os dados ou verificar alguma coisa;
3. Formato *Drag and Drop* para a criação de formulários é simples e rápido, sem necessidade de quaisquer outras ferramentas *HTML*;
4. *Emails* de notificação;
5. Capacidade de configurar uma resposta automática;
6. Enviar *emails* para qualquer número de endereços estáticos ou dinâmicos;
7. Controlo total sobre o modelo/estrutura de *email* a ser enviado;
8. Redireccionar os utilizadores para uma página de agradecimento ou outra página ou *site*;
9. Armazenar os dados submetidos e visualizá-los facilmente;
10. Capacidade de ver/apagar os dados armazenados e capacidade de cópia de segurança dos mesmos em formatos *XLS* ou *CSV*, visualizáveis em *MS Excel*;
11. Capacidade de criar tabelas de *BD* e se conectar a qualquer tabela para salvar ou actualizar os registos;
12. Incluir campos de *upload* de ficheiros e adicionar restrições sobre os ficheiros a serem enviados, como por exemplo o tipo de ficheiro e o tamanho mínimo e máximo;
13. Adicionar imagem de verificação *Captcha* de modo a evitar *Spam*;

14. Validação do lado do servidor e do cliente;
15. Fácil integração com sistemas de pagamento como o Paypal, 2Checkout.com ou Authorize.net;
16. Limite de envio de formulário por segundo por sessão de utilizador – útil para proteger quando o utilizador dá 2 cliques no botão Enviar;
17. Opção de DEBUG para verificar o fluxo de trabalho do formulário.

### B. *FormMaker*

É uma ferramenta que permite a construção de formulários de forma fácil, rápida e intuitiva sem ter de usar qualquer linguagem de programação. Permite definir parâmetros de um formulário, como cores e tipos de letra.

O *FormMaker* é ideal para utilizadores principiantes, mas também atende às necessidades de *designers* e especialistas. Pode ser usado para a criação de formulários de contacto, ficha de inscrição, convite para festa, etc.

[38]

Seguem-se algumas funcionalidades desta ferramenta: [38]

1. Compatível com todos os dispositivos móveis;
2. Totalmente integrado com o *Paypal*.;
3. Catorze elementos principais para a criação de um novo formulário;
4. Elemento HTML personalizado para a criação de um formulário;
5. Caixa de texto e de selecção, escolha múltipla e única, tempo e data (calendário);
6. *Upload* de ficheiros, quebra de página e de secção;
7. Mapa;
8. *Captcha*;
9. Botões (como por exemplo, botão de Enviar);
10. Múltiplas páginas de formulário;
11. Capacidade de separar o formulário em secções com número diferente de colunas;
12. Capacidade de usar *JavaScript* e *CSS*;
13. Opções de configuração de email avançadas;
14. Capacidade de criar itens de menu e se conectar com o formulário;
15. Capacidade para inserir um formulário num artigo Joomla;
16. Validação dos dados ao enviar o formulário;
17. Capacidade de armazenar os dados do formulário na base de dados;
18. Capacidade de restringir extensões de ficheiros para *upload*;
19. Integrado com a API do *Google Maps*.

### Ferramenta utilizada: *ChronoForms*

Foi escolhido o *ChronoForms* não só pela sua interface simples, intuitiva e flexível, como também pela sua capacidade de criar tabelas na BD e se conectar para salvar e/ou actualizar os dados introduzidos nos formulários. Além disso, o seu formato de *drag and drop* possibilita uma melhor organização dos campos durante a construção dos formulários.

Trabalhando num servidor local, a opção de *debug* é de grande utilidade pois permite simular o envio do formulário, mostrando também como seria o *email* enviado para o administrador, com os dados do formulário. O conteúdo desse *email*, pode ser estruturado pelo administrador, através das opções de configuração do formulário.

### 3.2.1.2. Drupal

#### A. *FormMaker*

O *FormMaker* disponível para Drupal é um módulo avançado destinado à criação de formulários. A interface de *back-end* do *FormMaker* é intuitiva e fácil de usar, permitindo que o utilizador crie formulários facilmente mesmo sem estar familiarizado com *scripts* e programação.

Este módulo satisfaz as exigências de utilizadores novatos e experientes, pois permite a criação de formulários simples e complexos. No entanto, se a complexidade do formulário requerer um mapa *Google*, o utilizador terá de comprar a versão comercial – ou optar por outra solução que não seja paga.

[39]

O *FormMaker* tem as seguintes características: [39]

1. Suporta o Drupal 6 e Drupal 7;
2. Integrado com o módulo *Views* do Drupal;
3. Personalização de HTML, CSS e *JavaScript* no formulário;
4. Capacidade de receber os formulários preenchidos no email definido (*email* do administrador);
5. Capacidade de restringir extensões de ficheiros no *upload* de ficheiros (disponível na versão comercial);
6. Validação de dados durante o envio do formulário;
7. Integrado com o editor *tinymce*, que permite editar facilmente o HTML;
8. Integrado com a API do *Google Maps* (disponível na versão comercial);
9. As mensagens de alerta do formulário podem ser facilmente traduzidas e alteradas.

#### B. *WebForm*

O *WebForm* é um módulo Drupal que permite criar formulários, tais como formulários de contactos, inquéritos, formulários de encomendas, reservas, etc. [40]

Algumas características do *WebForm* são:

1. Capacidade de receber os formulários preenchidos no email definido (email do administrador);
2. Os resultados podem ser exportados para o *Excel* ou outras aplicações semelhantes;
3. Permite atribuir e criar regras de validação de dados;
4. Permite introdução de HTML.

Ferramenta utilizada: *WebForm*

A solução escolhida foi o *WebForm*. Este tem uma interface simples, intuitiva e ainda possibilita a criação de regras de validação para os campos do formulário de forma simples, e a opção de receber por *email* dos dados introduzidos no formulário pelo utilizador. O conteúdo desse *email*, pode ser estruturado pelo administrador, através das opções de configuração no formulário. O seu formato de *drag and drop* é intuitivo e possibilita uma melhor organização dos campos durante a construção dos formulários.

### 3.2.2. Loja Virtual

Durante a primeira reunião com os *stakeholders* ficou claro que a existência de uma loja virtual era fundamental, pois desta forma poderiam alcançar mais público e usufruir de um novo meio para angariação de fundos.

Para a construção da mesma, no Joomla foram escolhidas as soluções *JoomShopping* e *VirtueMart*, e no Drupal o *Ubercart*.

#### 3.2.2.1. Joomla

##### A. *JoomShopping*

Disponível para Joomla 1.5, 1.6, 1.7, 2.5 e 3.0 – é uma solução de *e-commerce open source* que oferece uma série de características. É escrito em PHP e pode ser usado em ambientes típicos de PHP/MySQL. [41] [42] Com mais de 200 000 novos utilizadores em 3 anos o *Joomshopping* é considerado a Loja *Online* mais fácil de usar para Joomla. [41]

Este componente pode ser utilizado para a venda de diferentes produtos, tendo também funções para a venda de ficheiros de áudio ou vídeo. A descrição dos produtos pode ser apresentada com texto, imagens e também ficheiros de áudio ou vídeo. [41]

Algumas das características padrão deste componente são: [41] [43]

- ✓ Criação automática de uma factura personalizadas em PDF para os clientes (possível ajuste da óptica da factura);
- ✓ Módulos para fácil integração de pagamento aos fornecedores;
- ✓ Funcionamento simples e claro;

- ✓ Todas as funções típicas de uma loja *web*;
- ✓ Controlo de *stock* opcional;
- ✓ Código fonte limpo/simples para um ajuste rápido e boa classificação do motor de busca.

#### Funções Gerais [42]

1. Os clientes podem gerir as suas contas de utilizador;
2. Múltiplas moedas;
3. Vários métodos de pagamento (*Paypal*, dinheiro, débito na conta, transferência instantânea) e capacidade de adicionar novos métodos de pagamento;
4. Avaliação de produtos e comentários;
5. Multilíngue;
6. Gestão de número ilimitado de produtos e categorias. Imagens para todas as categorias;
7. Imagens, vídeos, ficheiros de som e demos podem ser adicionados à descrição de cada produto;
8. Podem ser adicionados vários atributos a cada produto;
9. Histórico de encomendas (o cliente pode ver todas as suas encomendas);
10. Lista de desejos (*wishlist*);
11. Gestão fiscal (cada produto pode ter o imposto diferente bem como as despesas de envio);
12. Gestão de entrega – múltiplas formas de entrega (transporte padrão, expresso); preço diferente consoante o país e o tamanho e peso da encomenda; capacidade de adicionar novos métodos de entrega;
13. O *email* de confirmação da encomenda é enviado para o cliente e para o administrador da loja;
14. Diferentes *templates* de categorias e produtos;
15. É possível comprar sem se registar.

#### Funções da administração [42]

1. Gestão de categorias e de produtos;
2. Lista de encomendas;
3. Gestão de fornecedores e de clientes;
4. Gestão de imposto;
5. Gestão de estado da encomenda (confirmado, cancelado, pendente, etc.);
6. Métodos de pagamento e de entrega (correio normal, expresso, etc.);
7. Preços de entrega;
8. Grupo de clientes para clientes (grupo VIP, grupo padrão, etc.);

9. Comentários;
10. Gestão de diferentes moedas e países;
11. Importar e exportar.

#### Requisitos mínimos [42]

1. Joomla 1.5.8
2. PHP – 4.3.10
3. MySQL – 4.0

#### **B. Virtuemart**

*VirtueMart* é uma extensão de loja virtual *open source* para o Joomla. Caracterizado como fácil de usar para novatos e especialistas, este oferece milhares de funcionalidades construídas para criar uma loja rapidamente. O *VirtueMart* permite ainda estender as suas funcionalidades por meio de *plugins*, módulos, *templates* e componentes.

A versão mais recente deste componente é compatível com o Joomla 2.5.

O *VirtueMart* oferece uma extensa lista de características padrão: [44]

1. Multilingue;
2. Categorias aninhadas;
3. Descrição do produto;
4. Vários tipos de ficheiros associados ao produto (imagens, vídeos, pdf, etc.);
5. *Feedback* (através do sistema de avaliação e comentários);
6. Opção de múltiplas moedas (Euro, Dolar, etc.);
7. Permite realizar compras sem registo;
8. Produtos relacionados;
9. Personalizar o formulário de registo de conta;
10. Poderosas funcionalidades de SEM/SEO;
11. Gestão de encomendas;
12. Gestão de oferta de cupões de desconto;
13. Adicionar novas funcionalidades à loja;
14. Motor de busca;
15. Gestão de clientes;
16. Controlo de *stock*;
17. Vários preços dependendo da moeda, grupo de cliente, etc.;
18. Dimensão dos produtos (tamanho e peso);
19. Atributos personalizados para produtos;
20. Vários métodos de pagamento e envio.

### Ferramenta utilizada: *JoomShopping*

De entre as duas ferramentas de loja virtual, o *JoomShopping* demonstrou-se ser o mais intuitivo para qualquer tipo de utilizador. Por esse motivo e por até à data da escrita deste documento, o *VirtueMart* não ser compatível com a versão **3.x** do Joomla, o *JoomShopping* foi a ferramenta escolhida. Relativamente ao carrinho de compras desta solução, este é muito mais simples e agradável do que o do *VirtueMart* que é um pouco confuso no que diz respeito à organização e remoção de produtos.

Uma falha a salientar tanto no *Joomshopping* como no *VirtueMart*, é a gestão de *stock* de produtos que tenham atributos associados, não permitindo introduzir a quantidade de *stock* por atributo. Por exemplo, ao criar um produto “camisola”, que tem associado o atributo “Cor” com os valores “Branco” e “Preto”, não é possível referir que existem 4 camisolas brancas e 2 pretas em *stock*, mas apenas que existem 6 camisolas em *stock*.

### 3.2.2.2. Drupal

#### A. *Ubercart*

*Ubercart* é uma loja virtual *open source* totalmente integrada com o Drupal [45]. A comunidade Drupal e *Ubercart* contribuem com *add-ons* que possibilitam a criação de lojas com funcionalidades de multilingue, cupões de desconto, carrinho de compras, diversos meios de pagamento *online*, atributos de produtos – como por exemplo a cor ou tamanho de um produto –, entre outros.

Seguem-se algumas das suas características: [45]

1. **Catálogo de produtos configurável** – inclui páginas de catálogo e um bloco para exibir categorias de produtos;
2. **Sistema flexível de criação de produto**. Criar produtos por padrão, i.e., produtos sem ficheiros de multimédia associados. Adicionar campos para guardar informações adicionais sobre o produto utilizando o sistema CCK (*Content Construction Kit*) do Drupal;
3. **Sistema flexível de atributos de produto** – permite criar atributos e associá-los a um ou mais produtos desejados, e ainda atribuir um preço a cada atributo. Este valor será depois adicionado ao valor base do produto;

Por exemplo:

Produto: Água = 1.00€

Atributos: Sabor | Maçã = 0.20€  
                          | Limão = 0.10€

Resultado: Uma água sem sabor custa apenas 1.00€ mas se for uma água de sabor a maçã irá custar 1.20€.

4. **Criação e edição simples de encomendas.** Fácil efectuar uma encomenda manualmente do lado do cliente, adicionando produtos, preço de envio, etc., a partir de um só ecrã;
5. **Sistema de pagamento integrado** que atua como uma ponte entre os métodos de pagamento aceitáveis (cheque, cartão de crédito, dinheiro, etc.) e *gateways* de pagamento (*Cyber Source, Authorize.net, PayPal*, entre outros);
6. **Registo de actividades.** Activar o registo para ver todas as alterações feitas a uma encomenda, incluindo processamento de pagamentos;
7. **Importação/exportação XML** permite importar/exportar produtos, atributos, encomendas, e os dados dos clientes da loja *Ubercart*.

Ferramenta utilizada: *Ubercart*

Durante a pesquisa de módulos de loja virtual para o Drupal, foi apenas encontrado o *Ubercart*, não se podendo excluir a hipótese de existirem outros. Sendo assim, a ferramenta escolhida foi o *Ubercart* não havendo uma outra que se pudesse comparar.

Contudo, é de salientar alguns pormenores em relação a esta ferramenta.

Durante a criação da loja virtual com o *Ubercart*, notou-se que esta solução de comércio electrónico de um modo geral, não é muito intuitiva. Ao activar o módulo *Ubercart*, previamente instalado, é criado um menu denominado *Store* no *dashboard* (Painel de Controlo). Este menu contém tudo o que é relacionado com a loja virtual, à excepção da criação de produtos. Para se criar um produto é necessário ir ao menu *Conteúdo* (presente no *dashboard*) e criar um conteúdo do tipo “*Product*” ou “*Product kit*”, o que não deixa de fazer sentido pois trata-se de um conteúdo, mas visto que essa opção é fundamental na criação de uma loja virtual, esta, a ver da autora deste documento, deveria estar, ou também estar, no menu *Store*, sendo este o menu referente à ferramenta de comércio electrónico.

Um outro aspecto relevante é a visualização do conteúdo do catálogo. Ao configurar o catálogo, para que este seja visível em determinada página/menu do *site*, não se obtém o efeito desejado. Sendo que para se poder visualizar o catálogo é necessário ir ao URL e acrescentar a palavra *catalog* (por exemplo: [http://localhost/nome\\_do\\_site/catalog](http://localhost/nome_do_site/catalog)), ou introduzir o bloco “*Categoria*” no *sidebar*, que apenas mostra o nome das categorias como um menu.

### 3.2.3. Calendário e Eventos

Ao concluir a implementação dos objectivos deste projecto, verificou-se a ausência de um método que possibilitasse a divulgação de eventos por parte do Centro da Mãe, desta forma foi adicionada a funcionalidade de calendário de eventos (*Events Calendars*), em resposta a esta necessidade.

#### 3.2.3.1. Joomla

##### A. *Really Simple Calendar*

O *Really Simple Calendar* é um componente de calendário simples para o Joomla, que permite a introdução de eventos associados a um ou mais dias do calendário. Este componente possibilita ao utilizador definir as cores do calendário, como por exemplo, a cor dos dias com eventos, a cor do dia actual, etc. [46]

##### B. *JEvents*

*JEvents* é um componente Joomla de calendário e eventos, traduzido em 37 línguas. Este componente permite criar categorias de eventos e eventos, que podem ser associados a uma ou mais categorias, e disponibiliza vários *layouts* de calendário para que o utilizador escolha aquele que mais lhe agrada. [47]

#### Ferramenta utilizada: *Really Simple Calendar*

A ferramenta eleita foi o *Really Simple Calendar* pela sua interface e facilidade de alterar as cores do calendário, pois o seu *back-end* disponibiliza uma opção para esse fim. O mesmo não acontece com o *JEvents*, onde só é possível modificar o aspecto do calendário, como por exemplo, a cor dos dias com eventos ou dia actual, através do(s) ficheiro(s) de CSS. Contudo, no que diz respeito à usabilidade o *JEvents* está mais bem organizado, dado que proporciona ao utilizador a visualização dos eventos, por dia, semana ou mês.

#### 3.2.3.2. Drupal

##### A. *Pretty Calendar*

É um módulo de calendário simples e com uma interface agradável que apresenta o conteúdo por data. A data pode ser seleccionada pela data de criação do nó (página) ou pelo campo de data fornecido pelo módulo “*Date*”. [48]

##### B. *Events Calendar Feature*

*Events Calendar Feature* é um módulo de calendário e eventos simples. Para usufruir desta funcionalidade é necessário seguir alguns passos descritos em [49], que vão desde a instalação de alguns módulos necessários, até à criação do bloco de calendário e exibição do mesmo, para a visualização dos eventos.

### Ferramenta Escolhida: *Events Calendar Feature*

Para a funcionalidade de calendário de eventos, a solução escolhida foi a *Events Calendar Feature* por ser a mais adequada para a divulgação de eventos.

Ao contrario da solução *Pretty Calendar*, que reconhece todos os artigos criados como sendo eventos – o que não é o pretendido –, o *Events Calendar Feature* permite a criação de conteúdo do tipo “eventos” que está associado ao calendário, por meio da vista criada, segundo o tutorial presente em [49]. Assim, o conteúdo visível no calendário, é apenas do tipo “eventos”.

No decorrer do processo de implementação do *Event Calendar Feature*, surgiram alguns contratemplos, visto que algumas configurações não surtiam o efeito desejado. Tendo em conta, não haver outro meio de configuração, e as mesmas estarem correctas, procedeu-se ao desabilitar do módulo e voltou-se a ligá-lo. Esta solução pode parecer trivial, mas resulta em algumas situações, e foi o caso aqui.

### 3.3. WEBSITE

Esta secção destina-se à apresentação das duas versões do *website* implementadas em Joomla e no Drupal e algumas observações referentes às mesmas.

Ao entrar no *website* o utilizador é direccionado para a *homepage* (menu **Início**) **Figura 18** e **Figura 19**, onde lhe são apresentado os artigos em destaque, seleccionados pelo administrador e o *banner* de publicidade (número 1 em ambas as figuras), à excepção do Drupal que conta com mais um elemento, o bloco de iniciar sessão (**Figura 19** número 2). Este bloco permite ao administrador aceder ao *back-end* – área de administração – e ao utilizador comum aceder à loja virtual, para efectuar as suas compras.

No menu **Quem Somos** e no seu submenu **Historial** é mostrado um artigo referente à história da associação Centro da Mãe e a sua cronologia, respectivamente.

Se o utilizador desejar efectuar donativos monetários ou em bens materiais, ou ser voluntário, poderá fazê-lo através do menu **Donativos**. Para ser voluntário ou sócio da associação é só preencher o formulário com os dados pedidos e submeter.

Para mais informações acerca da instituição existe o menu **Contactos**, que disponibiliza o horário de atendimento, a localização da associação Centro da Mãe e os seus vários meios de contactos disponíveis. Relativamente às imagens presentes neste menu, estas são chamadas através do URL por limitação do Drupal, que não suporta correctamente o código HTML uma vez que, introduzindo o caminho do directório estas ficam corrompidas. Por esta razão, a visualização dos ícones de imagens torna-se dependente do URL de um servidor externo.

Por fim, o menu **Loja Virtual**<sup>5</sup> permite que qualquer utilizador autenticado possa efectuar as suas compras *online*. Em relação a este menu, foi necessário realizar algumas alterações no código fonte do módulo/componente de loja virtual, no que diz respeito ao botão e *link* de “Comprar” e “Adicionar ao carrinho” de forma a que estes só fossem visíveis para utilizadores autenticados. Outro aspecto a salientar na Loja Virtual, é a apresentação deste menu no *website* em Drupal, que em vez de apresentar as categorias e os respectivos ícones de imagens no conteúdo da página – como na Loja Virtual do Joomla –, mostra um artigo com promoções referentes aos produtos, uma vez que o módulo de *e-commerce* Ubercart não permitia a visualização das categorias como ícones no conteúdo da página.

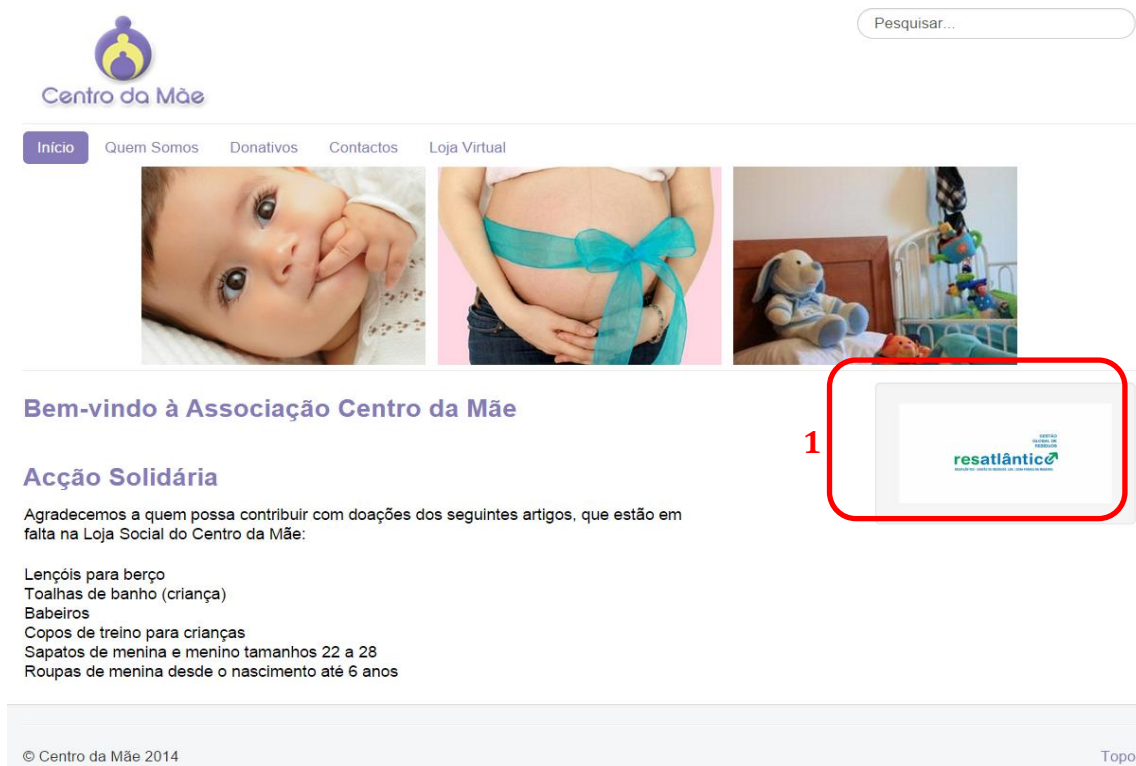


Figura 18- HomePage do website em Joomla

<sup>5</sup> **Loja Virtual** – os dados apresentados neste menu são apenas dados de exemplo, não correspondendo aos artigos realmente disponíveis para venda no Centro da Mãe.



Figura 19 - Homepage do website em Drupal

### 3.3.1. Configurando o SMTP

A fim de testar o *website* no seu todo, houve a necessidade de colocá-lo num servidor *online*. Neste caso foi colocado no domínio “*cee.uma.pt/cms*”. Assim, foi possível testar o serviço de SMTP em ambos os *websites* (Drupal e Joomla), ou seja, testar a recepção por *email* dos dados dos formulários de inscrição de sócio e voluntário.

Com isto, verificou-se que o serviço funcionava na perfeição com o *website* Drupal, o que não aconteceu no *website* Joomla. Visto que as configurações do serviço estavam iguais em ambos os *sites*, e num deles funcionava, o problema estaria então no Joomla.

Para resolver este problema, procedeu-se à pesquisa *online* de possíveis soluções, e verificou-se que possivelmente em algumas versões o Joomla teria problemas com o SMTP. No entanto, ao instalar as versões 3.2.7 e 3.3.5 sugeridas em [50] [51] o problema persistiu e até à data da escrita deste documento, não foi possível solucionar o problema.

## 4. DISCUSSÃO

Um CMS é benéfico para uma PME – neste caso uma IPSS – pois permite que esta tenha total autonomia sobre o conteúdo a ser divulgado e dispensa da assistência de empresas especializadas para manutenção, bem como de licenças do sistema por se tratar de um *software* livre, reduzindo assim os custos associados a estas tarefas, à excepção do custo de aprendizagem.

Por serem flexíveis e escaláveis, o *website* construído com um CMS cresce consoante a escalabilidade da empresa, i.e., consegue acompanhar as necessidades de expansão. No entanto, é importante mencionar que se um utilizador sem conhecimentos de programação, pretende criar algo que o sistema não disponibilize, deve preparar-se para ter que pagar algum programador, para que este possa desenvolver os conceitos desejados para o *website*. Outra alternativa seria dedicar-se ao estudo de linguagens de programação, pois as funcionalidades disponibilizadas para os CMSs podem nem sempre corresponder às expectativas.

Apesar de se poupar na gestão e manutenção, isto não implica que um *website* desenvolvido num CMS seja totalmente livre de custos, dependendo das necessidades da empresa, isto pode implicar alguns *plugins*/módulos pagos, pois nem todas as extensões dos CMS estão disponíveis gratuitamente, para além de outro custo inevitável que é o alojamento do *site*. Este último, pode variar consoante a empresa de alojamento, o serviço prestado pela entidade, e de outros critérios como, o espaço que o *site* ocupa em disco, o tráfego, etc. Em [52] [53] é possível visualizar os pacotes disponíveis por uma entidade de alojamento de *sites*, e os preços respectivos a cada pacote. Um exemplo de um pacote para este estudo de caso, seria o “Delta 0” para alojamento Linux ou “Libre 0” para alojamento Windows. Ambos os pacotes com 5Gb de espaço em disco, tráfego ilimitado, o envio de 150 *emails* por hora, BD ilimitada, entre outras características, tendo em conta tratar-se de um *site* de pequenas dimensões e com pouco tráfego.

Quanto aos requisitos em termos de *software*, que os CMS exigem – pelo menos os três estudados neste projecto – são:

- ✓ **Versão PHP:** 5.2+ (o símbolo “+” significa “ou superior”);
- ✓ **BD:** MySQL 5.1+;
- ✓ **Servidores Web Suportados:** Apache 2.0+, Nginx 1.0+
- ✓ **Browser:** Firefox 5+, IE 8+, Chrome 10+, Safari 5+.
- ✓ **Ambiente de desenvolvimento:** Windows, Linux, Mac OS.

Sendo que estes não exigem custos monetários, à excepção do sistema operativo. Em relação aos requisitos de *hardware* e custos associados, desde que se possua um

equipamento que suporte os requisitos de *software* anteriormente mencionados, o custo será nulo.

No que diz respeito às implicações da construção de um *website* pela forma tradicional e através de um CMS, é que este último, por ser o *software* e a base de dados a manter o controlo de cada parte do conteúdo do *site* (texto, *links*, menus, ficheiros multimédia, etc.), estes sistemas possuem um *back-end* (painel de administração) que permite adicionar recursos dinâmicos (mapas, carrinho de compras, etc.) na forma de *plugins*, directamente para o servidor. O utilizador pode mudar, adicionar ou remover o conteúdo quantas vezes quiser. Sendo a principal vantagem da utilização de um CMS, não necessitar de quase nenhum conhecimento técnico para gerir.

Por outro lado, um *site* desenvolvido pela forma tradicional é construído de raiz, através de um editor de linguagem de programação. Quando finalizado as páginas são enviadas para o servidor. Quaisquer alterações no conteúdo, menus ou *links* tem de ser feitas por meio do editor, o que pode ser um problema para os que não estão familiarizados ou não têm acesso ao servidor onde o *site* está hospedado.

Em suma, os Sistemas de Gestão de Conteúdo permitem que um *site* razoavelmente complexo seja criado com maior rapidez e baixo custo em relação a um *site* desenvolvido pela forma tradicional, partindo “do zero”.

Durante a implementação do *website* em Joomla e Drupal, foram feitos alguns testes de utilizador, como criar um artigo em cada CMS, e criar um produto na loja virtual em cada solução proposta para os respectivos CMS. Visto que os testes foram realizados apenas com um utilizador, não foram adquiridos dados em concreto, apenas o *feedback* pessoal do utilizador, que neste caso foi um colaborador do Centro da Mãe.

Assim, o *feedback* obtido durante os testes por parte do utilizador em relação à criação de artigos, foi que o Joomla seria o mais simples e intuitivo, pela sua interface administrativa e a interface de criação de artigos, por conter o editor de texto com elementos de formatação disponíveis, ao contrário do Drupal, onde, para formatar o conteúdo de um artigo, é necessário inserir *tags* HTML ou instalar um módulo de editor WYSIWYG. Relativamente à criação de um produto na loja virtual nas soluções propostas para o Joomla, o utilizador preferiu o *JoomShooping* ao invés do *VirtueMart*, pois tanto para a criação de um produto, como nas restantes funcionalidades de *back-end* (carrinho, etc.), esta solução apresentava-se mais simples e intuitiva, o mesmo acontecendo para o *front-end* das soluções propostas. No Drupal, sendo que o *Ubercart* é a única solução proposta, o *feedback* obtido foi de comparação em relação às soluções do Joomla, sendo as soluções deste último a sua preferência.

Em suma, para os testes de utilizador mencionados no parágrafo anterior e pela exploração em ambos os *websites* por parte do utilizador, o seu CMS de eleição seria o Joomla. Por fim, fazendo um balanço entre os dois CMS escolhidos para a implementação do *website*, pode-se concluir que o Joomla é muito mais amigável ao utilizador, recomendando-se assim, este CMS a qualquer tipo de utilizador, pois o seu nível de aprendizagem é acessível. Ao contrário do Drupal, que é um CMS direccionado para um tipo de utilizador mais específico, com conhecimentos de programação e/ou em desenvolvimento *Web*, dado que o nível de aprendizagem é mais exigente. Relativamente às suas comunidades, o Drupal dispõe de uma comunidade mais activa e que disponibiliza maior número e variedade de funcionalidades gratuitas em relação à comunidade Joomla. No entanto, das funcionalidades instaladas e estudadas, as do Joomla mostraram ser mais simples de usar, o que as torna amigáveis ao utilizador.



## 5. CONCLUSÃO

Neste trabalho, foi realizado o estudo comparativo dos CMS propostos e o desenvolvimento de um *website* em dois deles. A abordagem utilizada para realizar esses objectivos foi a análise individual de cada CMS, quanto às suas características, popularidade, complexidade, flexibilidade, escalabilidade, nível de aprendizagem (esforço) e qualidade do produto final, entre outros aspectos. Tendo concretizado os principais objectivos, foi acrescentado um elemento adicional, calendário de eventos, que seria útil e essencial para o *website* se tornar mais completo.

Os CMS – Sistemas de Gestão de Conteúdo – são sistemas relativamente novos no mercado e embora muitos não estejam familiarizados com eles, estes têm potencial de simplificar radicalmente a manutenção dos *sites*. Mas para tal também é necessário dedicar um certo tempo de estudo e análise para um perfeito entendimento e aproveitamento de todas as suas funcionalidades [54].

Depois do estudo de cada CMS, a opinião da autora é que para utilizadores com pouca experiência a melhor opção é o Joomla, uma vez que a sua flexibilidade, escalabilidade e facilidade de uso são de nível médio. Este CMS é ideal para *sites* um pouco complexos e de nível profissional, por exemplo, comércio electrónico, redes sociais, entre outros.

A sua interface de administração é intuitiva, simples e disponibiliza opções de formatação de texto, evitando assim a utilização de linguagem HTML que um utilizador comum e inexperiente não tem conhecimento.

A comunidade Joomla oferece extensões (*plugins*, componentes, etc.) comerciais e de domínio público para diversas categorias – como, comércio electrónico, gestão do *site*, mapas, calendários e eventos, entre outras – que contribuem para o enriquecimento do *website* a ser desenvolvido.

No caso de um utilizador com experiência e conhecimentos em programação e/ou em desenvolvimento *Web* o Drupal pode vir a ser o seu CMS de eleição uma vez que é ideal para *sites* complexos e profissionais, fornece mais flexibilidade – aptidão para diversos tipos de *site* – e escalabilidade.

A sua comunidade disponibiliza módulos e *templates* gratuitamente e um bom suporte de documentação. O tempo de resposta deste CMS é mais longo – no que diz respeito ao carregamento das páginas, principalmente quando estas possuem conteúdo de multimédia e/ou ficheiros de *JavaScript* – em relação aos outros CMS estudados.

Se os utilizadores são pouco experientes mas pretendem um *site* complexo, é imprescindível dedicação e estudo, pois o Drupal tem uma curva de aprendizagem mais acentuada.

Para utilizadores novatos e que pretendam *sites* simples, como blogues ou *sites* de revistas, o WordPress é uma boa solução. Oferece uma interface mais simples do que o Joomla e o Drupal, requer um nível de manutenção baixo porém, a sua flexibilidade e escalabilidade são limitadas. É também de salientar que apesar dos muitos *plugins* disponíveis, este CMS ainda se prende à estrutura de blogue para a qual foi desenvolvido inicialmente.

Com base no número de *downloads* de cada CMS observados nos respectivos *sites* [55] [56] [57] pode-se verificar que aquele que desperta mais interesse nos utilizadores é o WordPress. No entanto, pela preferência dos utilizadores, os valores apresentados em [55] [56] [57] não seriam os mesmos, como se pode verificar pela opinião da autora e pela informação presente na secção 2.5 (Comparação dos CMS), que apresenta a popularidade – desde o mais pesquisado, aos *sites* mais populares desenvolvidos com um CMS – e complexidades dos três CMS estudados, as suas vantagens e desvantagens, bem como alguns dados estatísticos presentes na ficha técnica.

## **TRABALHOS FUTUROS**

Os trabalhos futuros provêm de algumas limitações – pouca diversidade de módulos *e-commerce* e editores *WYSIWYG* desactualizados – encontradas no decorrer da implementação do *website*.

Assim propõe-se a implementação ou actualização dos editores de *WYSIWYG* para o Drupal 7 e respectivas versões posteriores, de modo a que a linguagem HTML funcione correctamente.

Propõe-se também a implementação de novos módulos de comércio electrónico para o Drupal, visto que durante a pesquisa deste só foi encontrado o *Ubercart*, que infelizmente possui algumas limitações, como a de apresentar o catálogo no menu pretendido pelo administrador, e é pouco intuitivo.

Relativamente aos *websites* desenvolvidos em Joomla e Drupal, uma ideia a complementar o *site* do Centro da Mãe, seria a funcionalidade de notificações para os utilizadores registados no *site*. Essas notificações iriam constar, por exemplo, em eventos agendados ou artigos disponíveis na loja virtual.

## REFERÊNCIAS

- [1] M. Raven, “CMS – Uma introdução aos Sistemas Gestores de Conteúdo Web,” 28 Novembro 2007. [Online]. Available: <http://www.hardware.com.br/artigos/cms/>. [Acedido em 25 Setembro 2014].
- [2] “A (brief) history of CMS development - Contegro,” 22 Janeiro 2012. [Online]. Available: [http://www.contegro.com/info-center/designers-blog/blog-article/\\_thread\\_/a-brief-history-of-cms-development](http://www.contegro.com/info-center/designers-blog/blog-article/_thread_/a-brief-history-of-cms-development). [Acedido em 25 Setembro 2014].
- [3] J. Robertson, “So, what is a CMS? - Step Two Designs,” 3 Junho 2003. [Online]. Available: [http://www.steptwo.com.au/papers/kmc\\_what/index.html](http://www.steptwo.com.au/papers/kmc_what/index.html). [Acedido em 22 Setembro 2014].
- [4] R. Shah, “Building a Web Content Management System,” Tese de Mestrado, San Diego State University, 2012.
- [5] The University of Kansas, “Benefits of the CMS | Content Management System,” [Online]. Available: <http://cms.drupal.ku.edu/benefits-cms>. [Acedido em 25 Setembro 2014].
- [6] Boney, “WordPress vs Joomla vs Drupal – CMS Popularity War,” 10 Janeiro 2013. [Online]. Available: <http://www.webnethosting.net/wordpress-vs-joomla-vs-drupal-cms-popularity-war/>. [Acedido em 8 Outubro 2013].
- [7] S. K. Patel, D. Rathod e J. B. Prajapati, “Performance Analysis of Content Management Systems- Joomla, Drupal and WordPress,” *International Journal of Computer Applications*, vol. 21, pp. 39-43, 2011.
- [8] D. Gwynn, “Open Source Web Content Management Systems in a Library Environment,” 7 Dezembro 2009. [Online]. Available: [http://davidgwynn.com/pdf/631\\_final.pdf](http://davidgwynn.com/pdf/631_final.pdf). [Acedido em 14 Outubro 2013].
- [9] M. Dexter e L. Landry, “Overview of Database Tables | How Joomla! Works | InformIT,” 12 Setembro 2011. [Online]. Available: <http://www.informit.com/articles/article.aspx?p=1728831&seqNum=9>. [Acedido em 18 Setembro 2014].
- [10] “Why Migrate - Joomla! Documentation,” 21 Fevereiro 2014. [Online]. Available: [http://docs.joomla.org/J3.2:Upgrading\\_from\\_an\\_existing\\_version](http://docs.joomla.org/J3.2:Upgrading_from_an_existing_version). [Acedido em 24 Fevereiro 2014].

- [11] “Plug-in - Wikipédia, a enciclopédia livre,” Wikipedia.org, 26 Março 2014. [Online]. Available: <http://pt.wikipedia.org/wiki/Plug-in>. [Acedido em 27 Março 2014].
- [12] “J3.x:Creating a Plugin for Joomla - Joomla! Documentation,” docs.joomla.org, 27 Agosto 2013. [Online]. Available: [http://docs.joomla.org/J3.2:Creating\\_a\\_Plugin\\_for\\_Joomla](http://docs.joomla.org/J3.2:Creating_a_Plugin_for_Joomla). [Acedido em 10 Março 2014].
- [13] “Drupal 7 - Mais fácil e mais poderoso do que nunca | Drupal.org,” drupal.org, [Online]. Available: <https://www.drupal.org/drupal-7.0/pt>. [Acedido em 30 Outubro 2014].
- [14] Crell, “Database API | Drupal.org,” drupal.org, 18 Setembro 2008. [Online]. Available: <https://drupal.org/developing/api/database>. [Acedido em 10 Dezembro 2013].
- [15] Drupal, “Update Drupal Core | Drupal.org,” drupal.org, 5 Maio 2013. [Online]. Available: <https://drupal.org/node/1494290>. [Acedido em 24 Fevereiro 2014].
- [16] H. Hafiz, “How to create custom module in drupal in three easy steps | drupal 6.x | webemania.com,” webemania.com, 28 Maio 2012. [Online]. Available: <http://www.webemania.com/blog/how-to-create-custom-module-in-drupal-in-three-easy-steps>. [Acedido em 10 Março 2014].
- [17] WordPress, “pt-br:Características do WordPress « WordPress Codex,” codex.wordpress.org, [Online]. Available: [http://codex.wordpress.org/pt-br:Caracter%C3%ADsticas\\_do\\_WordPress](http://codex.wordpress.org/pt-br:Caracter%C3%ADsticas_do_WordPress). [Acedido em 30 Outubro 2014].
- [18] WordPress, “Database Description « WordPress Codex,” codex.wordpress.org, [Online]. Available: [http://codex.wordpress.org/Database\\_Description](http://codex.wordpress.org/Database_Description). [Acedido em 10 Dezembro 2013].
- [19] “TRIPLEROI,” 2013. [Online]. Available: <http://www.tripleroi.com/cms/wordpress-development/default-mysql-database-tables>. [Acedido em 10 Dezembro 2013].
- [20] WordPress, “Updating WordPress « WordPress Codex,” codex.wordpress.org, [Online]. Available: [http://codex.wordpress.org/Updating\\_WordPress](http://codex.wordpress.org/Updating_WordPress). [Acedido em 24 Fevereiro 2014].
- [21] K. Mikoluk, “Drupal vs Joomla vs WordPress: CMS Showdown,” 18 Julho 2013. [Online]. Available: <https://www.udemy.com/blog/drupal-vs-joomla-vs-wordpress/>. [Acedido em 8 Outubro 2013].
- [22] Joomla, “Joomla! The CMS Trusted By Millions for their Websites,” [Online]. Available: <http://www.joomla.org/>. [Acedido em 8 Outubro 2013].

- [23] M. Mendes, "Youtube - "Joomla Tutorial Português Instalar Idioma PT Joomla! 1.5"," 8 Abril 2008. [Online]. Available: <http://www.youtube.com/watch?v=3HY3mTNd1iE>.
- [24] R. Silotto, "Youtube - "Joomla - Veja como Criar Formulários com o Componente ChronoForms"," 7 Agosto 2013. [Online]. Available: <http://www.youtube.com/watch?v=IROEc2rFHk0>.
- [25] Drupal, "ARCHIVE: Comparison of Drupal WYSIWYG Editors," 10 Fevereiro 2014. [Online]. Available: <https://www.drupal.org/node/208456>. [Acedido em 5 Abril 2014].
- [26] Joomla, "Joomla! 3.1.5 Stable Released," 1 Agosto 2013. [Online]. Available: <http://www.joomla.org/announcements/release-news/5505-joomla-3-1-5-stable-released.html>. [Acedido em 10 Dezembro 2013].
- [27] Joomla, "Release News," 26 Abril 2013. [Online]. Available: <http://www.joomla.org/announcements/release-news.html?start=12>. [Acedido em 10 Dezembro 2013].
- [28] Drupal, "Releases for Drupal core | Drupal.org," 5 Janeiro 2011. [Online]. Available: [https://drupal.org/node/3060/release?api\\_version\[0\]=103&page=1](https://drupal.org/node/3060/release?api_version[0]=103&page=1). [Acedido em 10 Fevereiro 2014].
- [29] Drupal, "What's New in Drupal 7 | Drupal.org," [Online]. Available: <https://drupal.org/about/new-in-drupal-7>. [Acedido em 10 Dezembro 2013].
- [30] WordPress, "Version 3.6 « WordPress Codex," [codex.wordpress.org](http://codex.wordpress.org/Version_3.6), [Online]. Available: [http://codex.wordpress.org/Version\\_3.6](http://codex.wordpress.org/Version_3.6). [Acedido em 24 Outubro 2014].
- [31] S. Balkhi, "What's New in WordPress 3.6," 1 Agosto 2013. [Online]. Available: <http://www.wpbeginner.com/news/whats-new-in-wordpress-3-6/>. [Acedido em 24 Outubro 2014].
- [32] WordPress, "Version 3.6.1 « WordPress Codex," [codex.wordpress.org](http://codex.wordpress.org/Version_3.6.1), [Online]. Available: [http://codex.wordpress.org/Version\\_3.6.1](http://codex.wordpress.org/Version_3.6.1). [Acedido em 10 Dezembro 2013].
- [33] H. Soares, "anossaescola.com," [Online]. Available: [http://www.anossaescola.com/cr/manuais/Joomla\\_tut\\_VisaoGeralTotos\\_v1.0\\_hugosoares2.pdf](http://www.anossaescola.com/cr/manuais/Joomla_tut_VisaoGeralTotos_v1.0_hugosoares2.pdf). [Acedido em 8 Janeiro 2014].
- [34] J. Carvalho, "Redes 11ºE Módulo 5: Joomla - Tipos de Utilizador - Redes 11ºE Módulo 5," 2 Abril

2011. [Online]. Available: <http://redesmodulo5.blogspot.pt/2011/04/joomla-tipos-de-utilizador.html>. [Acedido em 8 Janeiro 2014].
- [35] V. Teixeira, "THE GOOD ARTICLE," 30 Setembro 2011. [Online]. Available: <http://thegoodarticle.com/tipos-de-utilizadores-no-wordpress/>. [Acedido em 8 Janeiro 2014].
- [36] Chronoengine, "What is ChronoForms?," 12 Dezembro 2006. [Online]. Available: <http://www.chronoengine.com/18-news/latest/26-what-is-chronoforms.html>. [Acedido em 5 Maio 2014].
- [37] Joomla, "ChronoForms - The Joomla! Extensions Directory," [Online]. Available: <http://extensions.joomla.org/extensions/contacts-and-feedback/forms/1508>. [Acedido em 5 Maio 2014].
- [38] Joomla, "Form Maker - The Joomla! Extensions Directory," [Online]. Available: <http://extensions.joomla.org/extensions/contacts-and-feedback/forms/22319>. [Acedido em 5 Maio 2014].
- [39] Drupal, "Form Maker | Drupal.org," 8 Outubro 2012. [Online]. Available: <https://www.drupal.org/project/drupal-form>. [Acedido em 9 Abril 2014].
- [40] Drupal, "Webform | Drupal.org," 26 Abril 2004. [Online]. Available: <https://www.drupal.org/project/webform>. [Acedido em 9 Abril 2014].
- [41] Joomla, "JoomShopping - The Joomla! Extensions Directory," 9 Julho 2008. [Online]. Available: <http://extensions.joomla.org/extensions/e-commerce/shopping-cart/5378>. [Acedido em 9 Abril 2014].
- [42] "JoomShopping Funktionen," [Online]. Available: <http://www.webdesigner-profi.de/joomla-webdesign/joomla-shop/features.html?lang=en>. [Acedido em 9 Abril 2014].
- [43] "JoomShopping Your free online shop / shop component for Joomla," [Online]. Available: <http://www.webdesigner-profi.de/joomla-webdesign/joomla-shop.html?lang=en>. [Acedido em 9 Abril 2014].
- [44] VirtueMart, "VM Features," [Online]. Available: <http://www.virtuemart.net/features/all-features#>. [Acedido em 9 Abril 2014].
- [45] Ubercart, "What is Ubercart? | Ubercart," 29 Março 2012. [Online]. Available: [http://www.ubercart.org/what\\_is\\_ubercart](http://www.ubercart.org/what_is_ubercart). [Acedido em 30 Junho 2014].

- [46] Joomla, "Really Simple Calendar - The Joomla! Extensions Directory," [Online]. Available: <http://extensions.joomla.org/extensions/calendars-a-events/events/events-calendars/21623>. [Acedido em 31 Outubro 2014].
- [47] Joomla, "JEvents - The Joomla! Extensions Directory," [Online]. Available: <http://extensions.joomla.org/extensions/calendars-a-events/events/events-calendars/95?qh=YToyOntpOjA7czo3OiJqZXZlbnRzljtpOjE7czo5OiJqZXZlbnRzJ3MiO30%3D>. [Acedido em 2 Novembro 2014].
- [48] Drupal, "Pretty Calendar | Drupal.org," 18 Março 2012. [Online]. Available: [https://www.drupal.org/project/pretty\\_calendar](https://www.drupal.org/project/pretty_calendar). [Acedido em 3 Novembro 2014].
- [49] Drupal, "Step by step guide to create an event listing with calendar block in Drupal 7 | Drupal.org," 16 Agosto 2011. [Online]. Available: <https://www.drupal.org/node/1250714>. [Acedido em 3 Novembro 2014].
- [50] Joomla, "Joomla! - View Topic - Joomla 3.1.4 and SMTP connect failed problem.," 31 Julho 2013. [Online]. Available: <http://forum.joomla.org/viewtopic.php?f=706&t=811791>. [Acedido em 22 Setembro 2014].
- [51] "Can not send email - Grupos do Google," 2 Outubro 2014. [Online]. Available: <https://groups.google.com/forum/#!topic/joomla-dev-cms/c7KfIPi5v5M>. [Acedido em 6 Outubro 2014].
- [52] "Alojamento de Sites (Alojamento Web), Registo de Domínios, Revenda, Servidores Dedicados, Servidores Virtuais, Certificados SSL e Backups," [Online]. Available: <https://www.iberweb.pt/alojamento-de-sites/sharedhostinglinuxcompare.php>. [Acedido em 27 Outubro 2014].
- [53] "Alojamento de Sites (Alojamento Web), Registo de Domínios, Revenda, Servidores Dedicados, Servidores Virtuais, Certificados SSL e Backups," [Online]. Available: <https://www.iberweb.pt/alojamento-de-sites/sharedhostingwindowscompare.php>. [Acedido em 27 Outubro 2014].
- [54] T. F. L. de Medeiros, "http://www.petccufpb.com.br/wp-content/uploads/2008/05/CMS.pdf," 2008. [Online]. Available: <http://www.petccufpb.com.br/wp-content/uploads/2008/05/CMS.pdf>. [Acedido em 10 Outubro 2014].
- [55] "WordPress Download Counter," [Online]. Available: <https://wordpress.org/download/counter/>.

[Acedido em 3 Dezembro 2013].

[56] JoomlaCode, “JoomlaCode > Projects > Joomla > Releases > Browse Releases,” [Online]. Available: <http://joomlancode.org/gf/project/joomla/frs/>. [Acedido em 3 Dezembro 2013].

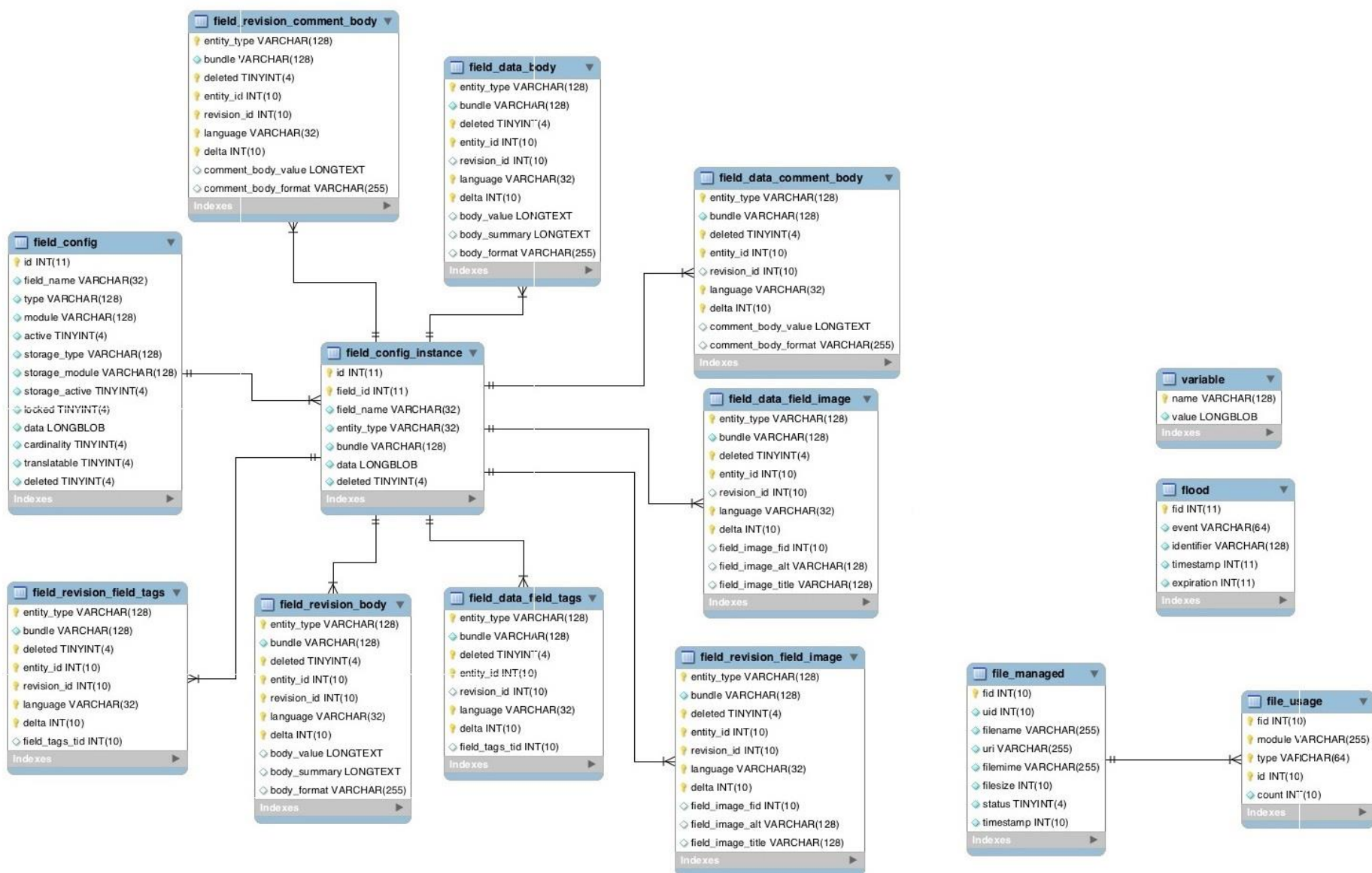
[57] Drupal, “Usage statistics for drupal 7.23 | Drupal.org,” [Online]. Available: <https://www.drupal.org/project/usage/2060023>. [Acedido em 3 Dezembro 2013].

[58] Wikipédia, “Cross-site scripting - Wikipédia, a enciclopédia livre,” 22 Outubro 2014. [Online]. Available: [http://pt.wikipedia.org/wiki/Cross-site\\_scripting](http://pt.wikipedia.org/wiki/Cross-site_scripting). [Acedido em 24 Outubro 2012].

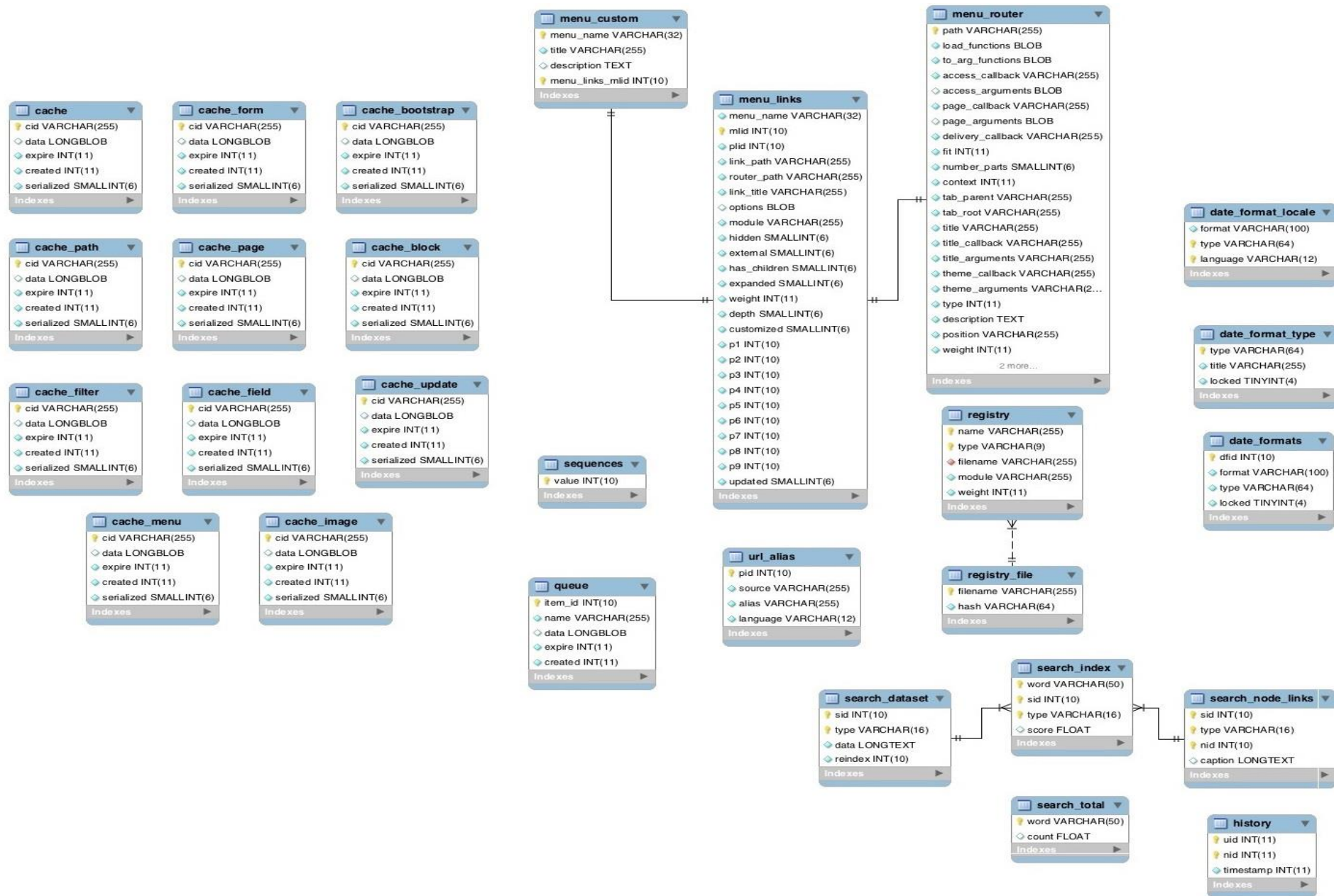
# ANEXOS



## Anexo 1 DIAGRAMA ENTIDADE-RELAÇÃO DO DRUPAL (PARTE 2) [14]



# Anexo 1 DIAGRAMA ENTIDADE-RELAÇÃO DO DRUPAL (PARTE 3) [14]



## Anexo 2 DIAGRAMA ENTIDADE-RELAÇÃO DO WORDPRESS [18]

