Projeto de Mestrado em Engenharia Informática

Computação Móvel

SGCPI

Sistema de Gestão de Conteúdos para Portais Institucionais

Cláudio Filipe Pedro Esperança

Leiria, março de 2015

Projeto de Mestrado em Engenharia Informática

Computação Móvel

SGCPI

Sistema de Gestão de Conteúdos para Portais Institucionais

Cláudio Filipe Pedro Esperança

Projeto de Mestrado realizado sob a orientação do Professor Doutor António Manuel de Jesus Pereira, Professor na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

Leiria, março de 2015

“Every success is built on the ability to do better than good enough”

(autor desconhecido)

Agradecimentos

A António Pereira, por aceitar o desafio de orientar este projeto e por todo o apoio, paciência e compreensão que demonstrou.

A Rita Cadima, por perceber a importância que deste projeto e ter criado as condições para a sua conclusão.

A Joana Mineiro, pelo apoio e recursos que forneceu para a realização deste projeto.

Aos meus fantásticos colegas da Unidade de Ensino a Distância do Instituto Politécnico de Leiria: Carina, Catarina, Joana, Manuela, Rita, Sandro, Sónia, Vítor, por todo apoio e ajuda.

A Marina Portugal, à família, amigos e colegas por toda a força, carinho e compreensão, mesmo quando o tempo não chega para tudo.

Resumo

Com o aumento exponencial da importância da Internet na sociedade, assumindo-se como um dos principais meios agregadores de informação da era digital, os portais Web são muitas vezes o primeiro ponto de contacto das instituições com as pessoas, servindo como um interface de comunicação aberto 24 horas por dia, 7 dias por semana. Deste modo, uma presença digital forte e apelativa torna-se fundamental para garantir que a instituição não fica em desvantagem competitiva relativamente aos seus concorrentes no meio digital.

O principal objetivo deste projeto era a construção de uma solução tecnológica para os portais Web para a disponibilização da nova imagem institucional do Instituto Politécnico de Leiria. Apesar do projeto ter como objetivo específico a construção de uma solução à medida com este fim, considera-se que os conceitos desenvolvidos no âmbito deste projeto podem ter um âmbito muito mais alargado que não o caso particular desta instituição.

Concretamente, este documento descreve o projeto de desenvolvimento, adaptação e personalização de uma solução tecnológica baseada em WordPress para a gestão de conteúdos para a Web, desde o desenho de uma proposta de arquitetura, à personalização gráfica da solução, implementação de alterações funcionais, desenvolvimento de módulos específicos, organização e gestão da informação, análise e otimização de código, entre outros temas.

Tecnicamente foram desenvolvidas algumas soluções inovadoras, tais como um mecanismo altamente modular para a implementação e isolamento de funcionalidades sobre um paradigma de programação orientada a eventos, um sistema virtual de modelos que permite implementar e reutilizar estruturas de páginas com um elevado grau de personalização, um mecanismo de associação de meta-campos aos vários tipos de conteúdos HTML suportados pela solução com interface amigável para inserção de informação e associação direta aos modelos virtuais e implementação de várias APIs específicas que permitem um elevado grau

de extensibilidade da solução.

Os resultados deste projeto traduzem-se na implementação de uma solução mais flexível e dinâmica, capaz de se adaptar aos cenários de utilização mais específicos de cada instituição. Ao basear o sistema numa plataforma bastante utilizada, reduz-se a curva de aprendizagem, permitindo que os gestores de conteúdo possam desempenhar o seu papel de forma mais simples e fácil.

Palavras-chave: Sistema de Gestão de Conteúdos para a Web, Portais institucionais, CMS, framework, WordPress

Abstract

With the increasing importance of the Internet in our society, assuming it as the main library of information in the digital age, an institution's website is often the first point of contact with their users, opened 24 hours a day, 7 days a week. Thus, a strong and appealing digital presence becomes critical to ensure that the institution is not at a competitive disadvantage with its competitors.

The main goal of this project was to implement a new CMS for the Polytechnic Institute of Leiria website as support for their new institutional image. Although the system have been design as a solution for the website of Polytechnic Institute of Leiria, we consider that the concepts developed during this project can be applied in other contexts other than the case of this particular institution.

This document describes the design and customization project of a WordPress CMS, from the proposal of a base system architecture, to the graphic customization, implementation of functional changes, development of specific modules, information management, code optimization, among other topics.

Technologically, some innovative solutions were implemented, such as a highly modular system to implement and isolate functionality over an event-driven programming paradigm, or a virtual template system on which page templates can be created and reused to implement content structures with a high degree of customization. A mechanism to create, manage and associate custom meta-fields to various types of HTML content supported by the system was also implemented, powered by friendly interface to easily insert and associate information to that content, which can be used directly on the virtual templates available in the system. Various specific APIs were also developed to further increase the extensibility of the solution.

As results of this project we have a more flexible and dynamic solution to Web content management, able to adapt to more specific usage scenarios of each institution. By using a widely used platform as the base system for the solution, the learning curve is reduced, helping content managers to do their work more easily and in a more simple way.

Keywords: Web Content Management System, CMS, Website, Framework, WordPress

Índice de figuras

Índice de quadros

Lista de siglas

Para referência deixamos uma lista de acrónimos e abreviaturas utilizadas ao longo do documento:

Acrónimos	Descrição das alterações
ADFS	*Active Directory Federation Services*
API	*Application Programming Interface*
CMS	*Content Management System*
CSS	*Cascading Style Sheets*
CSV	*Comma-separated values*
D3	*Design-driven development*
DEI	Departamento de Engenharia Informática
DER	Diagrama Entidade Relacionamento
DVCS	*Decentralized Version Control System*
ESTG	Escola Superior de Tecnologia e Gestão de Leiria
FDD	*Feature-driven development*
GPL	*GNU General Public License*
HTML	*Hyper Text Markup Language*
HTTP	*Hypertext Transfer Protocol*
IDE	*Integrated Development Environment*

Acrónimos	Descrição das alterações
IPL	Instituto Politécnico de Leiria
IPLeiria	Instituto Politécnico de Leiria
LAMP	Linux, Apache HTTP *server*, MySQL RDBMS *and* PHP
LDAP	*Lightweight Directory Access Protocol*
JS	JavaScript
MEI	Mestrado em Engenharia Informática
MEI-CM	Mestrado em Engenharia Informática – Computação Móvel
MVC	*Model–view–controller*
MVP	*Model–view–presenter*
NPM	*Node Package Manager*
PHP	PHP: *Hypertext Preprocessor*
RDBMS	*Relational Database Management System*
SASS	*Syntactically Awesome Stylesheets*
SEO	*Search Engine Optimization*
SGBD	Sistema de Gestão de Base de Dados
SGCPI	Sistema de Gestão de Conteúdos para Portais Institucionais
SQL	*Structured Query Language*
TinyMCE	*Tiny Moxiecode Content Editor*
UC	Unidade Curricular
UED	Unidade de Ensino a Distância

Acrónimos	Descrição das alterações
UI	*User Interface*
VCS	*Version Control System*
WAI-ARIA	*Accessible Rich Internet Applications*
WCAG	*Web Content Accessibility Guidelines*
WYSIWYG	*What You See Is What You Get*
XML	*Extensible Markup Language*
XSL	*Extensible Stylesheet Language*
XP	*Extreme Programming*

Índice

1. Introdução

Este relatório descreve as atividades realizadas pelo autor no âmbito do projeto intitulado SGCPI - Sistema de Gestão de Conteúdos para Portais Institucionais, tendo como objetivo o desenho e implementação de uma solução tecnológica de suporte à construção de portais institucionais, capaz de responder aos novos desafios e boas práticas da gestão de informação na Web. No sentido de avaliar a exequibilidade do sistema proposto no âmbito deste projeto, a solução foi implementada como plataforma de suporte aos novos portais institucionais do Instituto Politécnico de Leiria (IPLeiria) com as modificações necessárias para satisfazer as necessidades específicas da instituição. A linha de investigação associada a esta solução surge no âmbito do projeto de Mestrado em Engenharia Informática – Computação Móvel – da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, combinando os conhecimentos adquiridos ao longo das várias unidades curriculares que dão corpo a este curso, com conhecimentos específicos obtidos de uma análise mais aprofundada às tecnologias e boas práticas vigentes.

Nesta secção serão apresentados alguns dados estatísticos para contextualizar as motivações e objetivos gerais deste projeto, bem como os objetivos mais específicos relacionados com o caso particular do projeto de portais institucionais do IPLeiria. No final da secção descreve-se genericamente a estrutura do documento.

1.1. Motivação e objetivos

Atualmente é absolutamente indispensável que as instituições públicas e privadas tenham o seu espaço virtual na Internet. De acordo com estatísticas de 2014, existem mais de 3 mil

milhões de pessoas com acesso à Internet ou seja, cerca de 40% da população mundial [1]. Se compararmos este valor com as estimativas calculadas no ano 2000, assistimos a um crescimento na ordem dos 705% [2]. Em Portugal, estima-se que de mais de 7 milhões de pessoas têm acesso à Internet o que se traduz em aproximadamente 66% da população portuguesa [3].

Neste sentido, e num mercado cada vez mais eletrónico e global, uma presença na Internet torna-se essencial para garantir que os utilizadores têm acesso à informação sobre as instituições, os seus produtos e os seus serviços. Por outro lado, não basta ter um espaço na Internet para garantir o sucesso de uma estratégia comercial. Dado que este é, cada vez mais, um dos primeiros pontos de contacto com as instituições, é importante garantir que, a imagem é apelativa e que traduz a identidade da empresa, que o sistema é funcional e eficiente, e que tem atenção as caraterísticas dos visitantes (como por exemplo o idioma, questões de acessibilidade, etc.). Por outro lado, é também importante assegurar que o sistema se adequa visualmente e funcionalmente em função do dispositivo de acesso (*responsive web design* [4]), que facilita a indexação dos conteúdos por motores de pesquisa (*Search Engine Optimization* [5]), que a informação é dinâmica e está atualizada, entre muitos outros fatores associados ao sucesso de uma estratégia de comunicação na Web.

Por outro lado, num mercado em constante mutação, não é fácil acompanhar a constante evolução das tecnologias. Com este trabalho pretendeu-se desenvolver um sistema que pudesse evoluir ao longo do tempo em função das necessidades e do progresso tecnológico. Tentou-se evitar a implementação de uma solução demasiado fechada que possa restringir questões como atualizações do sistema (tanto de segurança, como funcionais), alterações gráficas e a evolução do próprio produto. Por outro lado tentou-se propor um sistema onde o seu potencial e extensibilidade não comprometesse a sua usabilidade, garantindo que se possa implementar um sistema que seja intuitivo para o público a que se destina, tornando a tecnologia transparente e centrando o sistema nas necessidades do utilizador.

Para o IPLeiria este projeto devia permitir adequar a sua presença na Web às novas realidades do mercado, de acordo com os seguintes objetivos mais específicos:

- Modernização da imagem da instituição na Web;
- Melhoraria da usabilidade e facilitar o acesso à informação;
- Adequação da informação aos objetivos estratégicos da instituição e ao público-alvo;
- Disponibilização de conteúdos em vários idiomas;
- Adicionar suporte à navegação com dispositivos móveis;
- Melhoria da acessibilidade dos portais.

1.2. Estrutura do documento

Após esta introdução, começaremos por apresentar - numa visão mais global - o estado da arte e - numa perspetiva mais particular – a situação inicial do IPLeiria. Depois serão apresentadas as especificações do projeto, nomeadamente os requisitos da solução e arquitetura geral da solução.

Na secção 4 será discutida a implementação da solução, onde descreveremos a equipa de projeto, o plano de projeto, as ferramentas de desenvolvimento utilizadas, a estrutura dos temas adotada, os módulos globais desenvolvidos, bem como os temas e módulos específicos desenvolvidos.

Na secção 5 serão descritos os procedimentos gerais para colocação da solução em produção, será descrita a arquitetura da infraestrutura de produção do ambiente de produção, a configuração do ambiente e parametrização da solução, a inserção dos conteúdos iniciais, a apresentação do protótipo à comunidade, a formação e apoio à gestão de conteúdos, e por fim, a disponibilização da solução ao público.

Concluiremos o documento com os resultados e conclusões do projeto, com a erro: origem da

referência não encontrada do documento e os vários erro: origem da referência não encontrada que complementam a informação sobre este projeto.

2. Estado da arte

Esta secção apresenta uma visão mais global sobre os Sistemas de Gestão de Conteúdo (CMS) que suportam o Web, onde a análise dos dados apresentados confirma que o CMS mais utilizado atualmente é o WordPress. Na segunda parte, apresenta-se o caso do IPLeiria, nomeando algumas das tecnologias utilizadas na instituição para a criação e gestão de conteúdos na Web, tentando contextualizar o cenário que levou à transição dos sistemas anteriores para esta nova plataforma.

2.1. Visão global

A empresa Netcraft identificou em Fevereiro de 2015, 883.419.935 *sites* de Internet [6], suportados por milhares de plataformas de gestão de conteúdo (CMS - *Content Management Systems*). Apesar de não ser obrigatório a utilização de um sistema de gestão de conteúdos para a disponibilização e gestão da informação de um espaço na Internet, a utilização deste tipo de tecnologia permite simplificar o processo de gestão de conteúdos, permitindo que utilizadores com um perfil menos técnico possam assumir o papel gestores de informação de um *site*. Este é um dos principais motivos pelos quais a maioria dos projetos Web de maior de dimensão acaba por escolher sistemas deste tipo.

Entre soluções *open-source*, soluções proprietárias, ou outras soluções à medida, estão disponíveis no mercado muitas plataformas CMS. Assumindo a métrica dos 10 milhões de *sites* mais visitados na Web (com base na classificação da empresa Alexa e análise estatística da empresa W³Techs) [7], 23% de todos estes *sites* Web utilizam a plataforma WordPress[1]

1 https://wordpress.org/

como solução tecnológica para a gestão de conteúdos dos seus *sites* [8], 2,9% utilizam a plataforma Joomla![2], 2,0% utilizam o Drupal[3] e, para referência, 0.1% deste grupo de *sites* optou pela plataforma SharePoint[4] da Microsoft. Se assumirmos os números publicados pelo serviço Built With para os 10.000 *sites* mais populares, 48,02% dos sistemas monitorizados (e que utilizam tecnologias CMS) adotaram o WordPress como a sua solução para a gestão de conteúdos Web, 14,86% escolheram o Drupal, 1,13% o Joomla!, não tendo sido encontrado um valor de referência para a plataforma SharePoint nesta estimativa [9].

Mesmo sem números exatos, e aceitando as variações estatísticas associadas às métricas utilizadas para elaboração das estimativas de referência, analisando estes resultados é possível identificar que a plataforma WordPress é o CMS mais utilizado. Verifica-se também que as três soluções mais utilizadas são *open-source*, o que parece indicar uma tendência para este tipo de sistemas.

2.2. O caso do IPLeiria

Ao longo dos anos, o IPLeiria tem utilizado vários CMS como soluções tecnológicas para os seus portais Web. Em algumas escolas foi utilizada a plataforma comercial "SWP Portal" com alguns módulos personalizados desenvolvidos pela Sitework/inCentea. Durante alguns anos o portal do IPLeiria foi também suportado por um CMS desenvolvido à medida por uma equipa de técnicos da instituição. Mais tarde, o IPLeiria adotou para o seu portal e para os portais das suas escolas uma solução tecnológica baseada em SharePoint. Ao longo de 7 anos de desenvolvimento, esta solução passou por várias iterações no sentido de tentar acompanhar as várias alterações que foram sendo solicitadas durante o tempo de vida do projeto. No entanto este sistema continuou a apresentar problemas de usabilidade, de acessibilidade e técnicos (como por exemplo, na pesquisa, gestão do fim de sessão, apresentação de opções não

2 http://www.joomla.org/

3 https://www.drupal.org/

4 https://products.office.com/en-us/sharepoint/

adequadas ao perfil do utilizador, etc). No que diz respeito à gestão da informação, os gestores de conteúdos queixavam-se da complexidade da solução, falta de flexibilidade e de controlo, desempenho e aspeto visual desatualizado.

Figura 2.1: Portal do IPLeiria baseado em SharePoint

Paralelamente a este projeto, face à grande quantidade de *micro-sites* solicitados à Unidade de

SGCPI

Ensino a Distância (UED) do IPLeiria para projetos como, apresentação de grupos de investigação, disponibilização de eventos, divulgação de projetos, etc., os técnicos desta unidade implementaram um sistema *multi-site* baseado em WordPress[5], como um sistema integrado onde foram implementados várias dezenas de *sites* sob um único sistema partilhado por todos os projetos.

Figura 2.2: Plataforma multi-site do IPLeiria baseada em WordPress

5 Esta solução pode ser consultada no endereço: http://sites.ipleiria.pt/

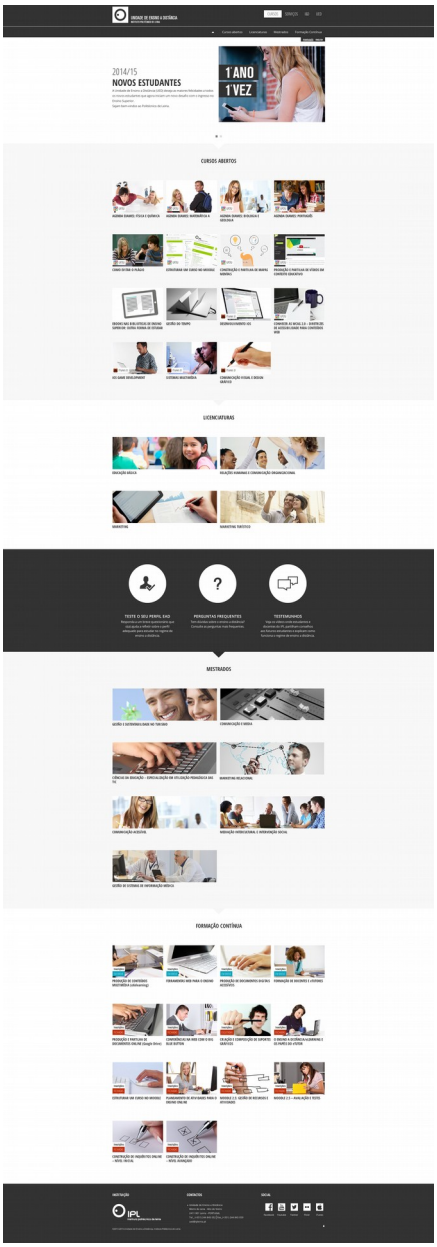


Figura 2.3: Página da UED no sistema multi-site

Esta solução permitia criar um *site* Web, associar utilizadores ao mesmo a partir do domínio do IPLeiria, definir e configurar um tema e disponibilizar o *site* ao público em poucos minutos. No caso de ser solicitado uma identidade gráfica própria, depois de recebida a proposta de *layout* devidamente aprovada pelo requerente do serviço, era possível criar um novo *site* com um aspeto próprio ao projeto num intervalo de tempo que poderia variar entre algumas horas e alguns dias, dependendo da complexidade do sistema a implementar. Para os utilizadores da plataforma a perceção na utilização de cada *site* é que este se trata de um sistema isolado, não tendo qualquer indicação que estão efetivamente a utilizar um ambiente partilhado com centenas de outros utilizadores.

Atualmente esta solução suporta cerca de 200 sites, tem cerca de 2250 utilizadores registados no sistema, utiliza aproximadamente 130 temas distintos, completados por 109 extensões. Esta infraestrutura é suportada por um único servidor, com caraterísticas semelhantes às de um computador pessoal.

Face às críticas e problemas identificados na solução baseada em SharePoint e ao sucesso da solução de *sites* baseada em WordPress, a presidência do IPLeiria propôs à UED o desenvolvimento de um novo sistema que apresente uma solução aos problemas identificados, que assumem a forma dos requisitos propostos na próxima secção.

3. Especificações

Nesta secção são apresentados os principais requisitos (e respetiva terminologia) identificados ao longo do ciclo de desenvolvimento do projeto e que traduzem as necessidades específicas do projeto de portais institucionais do IPLeiria. Na segunda parte são apresentadas duas propostas para uma arquitetura geral da solução baseada num paradigma de programação orientada a eventos que permita a criação de um sistema altamente modular. Serão abordadas questões como níveis de acoplamento e interdependência entre módulos, será explicado o funcionamento de um sistema que siga um paradigma de programação orientada a eventos e serão apresentadas duas propostas de arquitetura semelhantes, onde a principal diferença reside na utilização, ou não, de uma plataforma existente para sustentação da solução.

3.1. Requisitos da solução

Nesta secção serão identificados os principais requisitos identificados como necessidades para as quais esta solução devia apresentar uma resposta. Os requisitos apresentados não foram todos identificados na fase inicial do projeto, dado que à medida que se ia tomando conhecimento da dimensão do problema, eram identificadas novas necessidades que tinham de ser inseridas na solução. Neste sentido o sistema proposto teve de ser suficientemente flexível para permitir a implementação de uma qualquer funcionalidade específica não contemplada na identificação de requisitos inicial, tentando minimizar o impacto destes novos requisitos na duração do projeto.

Esta secção está dividida um sub-secções onde iremos apresentar a terminologia utilizada nos requisitos, sendo depois identificados os requisitos funcionais (prefixo RF), não funcionais

(RNF) e de desenvolvimento da solução (RD), com a indicação da respetiva prioridade de implementação.

3.1.1. Terminologia

Para introduzir os termos utilizados e facilitar a compreensão dos requisitos, na Tabela 1 desta secção apresentamos a terminologia utilizada.

Termo	Descrição
Visitante	Perfil do utilizador anónimo que visita o *site* mas não tem sessão iniciada no sistema.
Utilizador autenticado	Perfil do utilizador com sessão iniciada no *site*, que se autenticou no sistema e é reconhecido pelo mesmo como sendo um utilizador específico.
Assinante	*Utilizador autenticad*o, com acesso à sua informação de perfil.
Colaborador	*Assinante*, que pode produzir determinados tipo de conteúdos no sistema, embora não os possa disponibilizar publicamente.
Autor	*Colaborador*, que pode criar, editar e publicar os conteúdos da sua autoria.
Editor	*Autor*, que pode gerir (criar, editar, publicar, eliminar, agendar, etc.) os seus conteúdos e os de outros utilizadores.
Administrador	*Editor*, que pode também administrar definições e opções administrativas.
Super-administrador	*Utilizador autenticado*, com privilégios de administrador em qualquer *site* da rede que faça parte do sistema.
Publicar	Disponibilizar um conteúdo no sistema a visitantes.
Página	Recurso com conteúdo HTML que pode ser apresentado aos utilizadores.

Termo	Descrição
Artigo	Notícia com conteúdo HTML que pode ser apresentado aos utilizadores.
Gestão	Processo de criar, editar ou eliminar um recurso do sistema.

Tabela 1: Terminologia utilizada nos requisitos

3.1.2. Requisitos funcionais

Nesta secção são descritos os principais requisitos funcionais da solução, sendo identificados (sempre que possível), a **ação**, o *autor* da ação e os objetos influenciados/envolvidos na ação. Esta lista de requisitos não é exaustiva e serve apenas de referência para identificar os principais requisitos do projeto e se ter uma ideia do tipo de necessidades que a solução proposta tem capacidade para responder.

Referência	Descrição	Prioridade
RF001	O sistema deve permitir ao *Editor* **inserir** páginas.	Alta
RF002	O sistema deve permitir ao *Editor* **editar** páginas de todos os *Editores*.	Alta
RF003	O sistema deve permitir ao *Editor* **eliminar** páginas de todos os *Editores*.	Alta
RF004	O sistema deve permitir ao *Editor* **publicar** páginas de todos os *Editores*.	Alta
RF005	O sistema deve permitir ao *Editor* **inserir** artigos.	Alta
RF006	O sistema deve permitir ao *Editor* **editar** artigos de todos os *Colaboradores*.	Alta
RF007	O sistema deve permitir ao *Editor* **eliminar** artigos de todos os *Colaboradores*.	Alta

SGCPI

Referência	Descrição	Prioridade
RF008	O sistema deve permitir ao *Editor* **publicar** artigos de todos os *Colaboradores*.	Alta
RF009	O sistema deve permitir ao *Autor* **inserir** artigos.	Alta
RF010	O sistema deve permitir ao *Autor* **editar** artigos da sua autoria.	Alta
RF011	O sistema deve permitir ao *Autor* **eliminar** artigos da sua autoria.	Alta
RF012	O sistema deve permitir ao *Autor* **publicar** artigos da sua autoria.	Alta
RF009	O sistema deve permitir ao *Colaborador* **inserir** artigos.	Alta
RF010	O sistema deve permitir ao *Colaborador* **editar** artigos da sua autoria.	Alta
RF011	O sistema deve permitir ao *Editor* **alterar** o estado dos artigos de todos os colaboradores.	Alta
RF012	O sistema deve permitir ao *Autor* **alterar** o estado dos artigos da sua autoria.	Alta
RF013	O sistema deve permitir ao Colaborador a **associação** de categorias a artigos.	Média
RF014	O sistema deve permitir ao *Editor* **inserir** categorias de artigos.	Média
RF015	O sistema deve permitir ao *Editor* **editar** categorias de artigos.	Média
RF016	O sistema deve permitir ao *Editor* **eliminar** categorias de artigos.	Média
RF017	O sistema deve permitir ao *Super-administrador* a **gestão** de *sites*.	Alta
RF018	O sistema deve permitir ao *Administrador* a **associação** de utilizadores a *sites* com um determinado perfil.	Alta

Referência	Descrição	Prioridade
RF019	O sistema deve permitir *Visitante* **iniciar sessão** com as suas credenciais do IPLeiria (autenticação LDAP).	Média
RF020	O sistema deve permitir ao *Super-administrador* a **gestão** de contas de utilizador.	Alta
RF021	O sistema deve permitir ao *Visitante* a **seleção** de idioma.	Alta
RF022	O sistema deve permitir ao *Administrador* a **gestão** de idiomas dos *sites* que administra.	Média
RF023	O sistema deve permitir aos *Utilizadores autenticados* a **gestão** dos conteúdos sob os quais têm privilégios nos vários idiomas disponíveis para os respetivos *sites*.	Média
RF024	O *Super-administrador* deve ter, pelo menos, os mesmos privilégios que um *Administrador* em cada *site*.	Alta
RF025	O *Administrador* deve ter, pelo menos, os mesmos privilégios que um *Editor* para um mesmo *site*.	Alta
RF026	O *Editor* deve ter, pelo menos, os mesmos privilégios que um *Autor* para um mesmo *site*.	Alta
RF027	O *Autor* deve ter, pelo menos, os mesmos privilégios que um *Colaborador* para um mesmo *site*.	Alta
RF028	O *Colaborador* deve ter, pelo menos, os mesmos privilégios que um *Assinante* para um mesmo *site*.	Alta
RF029	O sistema deve permitir ao *Editor* a **gestão** de testemunhos dos estudantes.	Média

SGCPI

Referência	Descrição	Prioridade
RF030	O sistema deve permitir ao *Editor* a **gestão** da oferta formativa do IPLeiria.	Alta
RF031	O sistema deve permitir ao *Editor* a **categorização** da oferta formativa do IPLeiria por Áreas Científicas.	Alta
RF032	O sistema deve permitir ao *Editor* a **categorização** da oferta formativa do IPLeiria por Escolas.	Alta
RF033	O sistema deve permitir ao *Editor* a **categorização** da oferta formativa do IPLeiria por Regimes.	Alta
RF034	O sistema deve permitir ao *Editor* a **categorização** da oferta formativa do IPLeiria por Idiomas.	Alta
RF035	O sistema deve permitir ao *Editor* a **categorização** da oferta formativa do IPLeiria por Tipos de formação.	Alta
RF036	O sistema deve permitir ao *Visitante* a **aplicação** de filtros na oferta formativa por Escola, Área cientifica, Regime, Idioma e Tipo de formação.	Alta
RF037	O sistema deve permitir ao *Visitante* a **pesquisa** de oferta formativa do IPLeiria.	Baixa
RF038	O sistema deve permitir ao *Editor* a **gestão** dos protocolos (descontos) para a comunidade do IPLeiria.	Alta
RF039	O sistema deve permitir ao *Editor* a **categorização** dos protocolos do IPLeiria por Tipos.	Alta
RF040	O sistema deve permitir ao *Visitante* a **aplicação** de filtros nos protocolos por Tipo.	Alta

Referência	Descrição	Prioridade
RF041	O sistema deve implementar um mecanismo para a apresentação de imagens em janelas modais.	Baixa
RF042	O sistema deve fornecer um mecanismo para a **gestão** e **reutilização** de modelos de conteúdo.	Média
RF043	O sistema deve fornecer um mecanismo para a **criação** de *slideshows* na área de conteúdo.	Baixa
RF044	O sistema deve fornecer um sistema para **compressão** ou **expansão** de zonas de conteúdo.	Média
RF045	O sistema deve fornecer uma *framework* CSS para facilitar a **criação de conteúdos** *responsive*.	Alta
RF046	O sistema deve permitir a **criação** de galerias de imagens no conteúdo.	Baixa
RF047	O sistema deve fornecer um mecanismo de **definição** centralizada de uma lista de ligações úteis.	Alta
RF048	O sistema deve fornecer um mecanismo de **definição** centralizada do rodapé institucional.	Alta
RF049	O sistema deve permitir a **incorporação** do rodapé institucional em *sites* de terceiros.	Média
RF050	O sistema deve permitir a **incorporação** da lista de ligações úteis em *sites* de terceiros.	Média
RF051	O sistema deve permitir a **associação** de menus à área de conteúdo.	Média
RF052	O sistema deve fornecer um mecanismo para **seleção** do idioma do sistema pelo *visitante*.	Alta

SGCPI

Referência	Descrição	Prioridade
RF053	O sistema deve fornecer um mecanismo para **detetar automaticamente** a lista de idiomas preferidos do *visitante* da lista de idiomas do navegador e **apresentar** os conteúdos do sistema num desses idiomas (se disponível).	Baixa
RF054	O sistema deve fornecer um mecanismo de **redirecionamento** de pedidos no formato do sistema antigo, para uma cópia do sistema antigo.	Média
RF055	O sistema deve fornecer um mecanismo para a **gestão**, **associação** e **apresentação** de campos de dados a conteúdos no sistema.	Média
RF056	O sistema deve fornecer um mecanismo de **sincronização** de menus entre os vários *sites* que compõem o sistema.	Baixa
RF057	O sistema deve fornecer um mecanismo para **seleção** das imagens de conteúdo adequadas em função das caraterísticas do *dispositivo de acesso*.	Baixa

Tabela 2: Requisitos funcionais

3.1.3. Requisitos não-funcionais

Nesta secção são apresentados os principais requisitos não-funcionais e de âmbito mais genérico da solução. Preocupações com questões de usabilidade, de aparência entre outras, são abordadas nesta secção.

Referência	Descrição	Prioridade
RNF001	Deve existir uma consistência visual entre os vários *sites* do sistema, permitindo uma transição transparente entre os sites que assumem a identidade gráfica do projeto.	Alta

Referência	Descrição	Prioridade
RNF002	O sistema deve ser simples e intuitivo.	Alta
RNF003	O sistema deve disponibilizar uma interface para a apresentação da oferta formativa do IPLeiria.	Alta
RNF004	O sistema deve estar preparado para apresentar conteúdos em vários idiomas (numa primeira fase, o Português e o Inglês).	Alta
RNF005	A implementação deve seguir as propostas gráficas do projeto e aprovadas pela direção.	Alta
RNF006	O sistema deve fornecer um mecanismo flexível para a gestão dos campos associados à oferta formativa do IPLeiria.	Alta
RNF007	Deve ser respeitada a regra do 3 cliques, onde uma qualquer informação relevante deverá estar acessível com apenas 3 interações.	Baixa
RNF008	O sistema deve ser flexível para se adaptar em função das necessidades ao longo do tempo de vida do projeto.	Baixa
RNF009	O sistema deve ser rápido e reativo.	Médio
RNF010	O sistema deve ter em consideração técnicas de *responsive design* no sentido de garantir um comportamento adequado em função dos dispositivos de acesso.	Alta
RNF011	O sistema deve ser conforme com as normas de acessibilidade vigentes, nomeadamente as WCAG 2.0 [10] e WAI-ARIA 1.0 [11].	Alta
RNF012	O sistema deve ser graficamente flexível, permitindo modificar completamente a estrutura e aparência em páginas diferentes de acordo com as necessidades identificadas para esse conteúdo.	Alta

SGCPI

Referência	Descrição	Prioridade
RNF013	O sistema deve ser disponibilizado no mais curto espaço de tempo possível.	Média

Tabela 3: Requisitos não-funcionais

3.1.4. Requisitos de desenvolvimento

No âmbito deste projeto não foram impostos quaisquer requisitos de desenvolvimento e foi dada total autonomia para apresentação de uma proposta que fosse considerada a mais adequada para a solução.

3.2. Arquitetura geral

Para este tipo de projeto entendeu-se que a solução deve assumir uma arquitetura modular, que permita encapsular funcionalidades específicas em módulos com um baixo nível de acoplamento [12]. Ao isolar a funcionalidade em módulos e reduzir o seu nível de interdependência, estamos a criar as condições para a paralelização do processo de desenvolvimento (onde programadores diferentes podem trabalhar em funcionalidades diferentes), garantindo também que cada módulo possa evoluir de forma isolada em função das necessidades e objetivos do projeto. Apesar de se privilegiar um baixo nível de acoplamento, dependendo do grau de relação de determinadas funcionalidades, poderá fazer sentido que exista uma comunicação direta entre os módulos que encapsulam estas funcionalidades.

3.2.1. Programação orientada a eventos

Esta arquitetura recomenda ainda que se equacione a adoção de um paradigma de

programação orientado a eventos, para troca de informações entre os vários módulos da solução e gestão do fluxo de execução do *software* para cada pedido recebido. Segundo este paradigma, numa primeira fase do fluxo de execução de um programa, funções e/ou métodos registam-se como "ouvintes" de determinada ação, evento ou filtro (*hooks*). Depois, ao longo do fluxo de execução do programa, estas ações ou eventos são despoletados permitindo que as funções registadas como "ouvintes" sejam executadas nos momentos adequados. Este modelo facilita o processo de modularização da arquitetura, delegando nos módulos a responsabilidade de se associarem aos eventos ou ações (reconhecidas pelo sistema) para concretizarem o comportamento específico pretendido. Assim o sistema não tem de conhecer a arquitetura de cada um dos módulos ou apresentar uma API rígida sob a qual a funcionalidade dos módulos tenha de ser implementada. Desta forma, o sistema apenas necessita de fornecer os *hooks* aos quais o comportamento dos módulos vai ser associado. Dado que as funcionalidades estão encapsuladas em módulos, esta arquitetura permite que os módulos possam ser removidos sem comprometer o funcionamento geral da solução, ignorando a perda de funcionalidade associada a esses módulos.

Ao permitir a associação de múltiplos "ouvintes" a um mesmo filtro, estamos ainda a permitir a implementação de um possível comportamento em cascata, onde o sistema pode fornecer um conteúdo inicial associado à execução desse evento, que vai sendo modificado por cada uma das funções ou métodos associados a esse mesmo *hook*, filtrando ou modificando o conteúdo de acordo com o comportamento pretendido. Por exemplo, podemos ter um *hook* responsável por filtrar o conteúdo de uma página HTML, onde através de expressões regulares cada módulo pode substituir uma parte do conteúdo fornecido pelo sistema, pelo código final com a concretização da funcionalidade específica que, após a conclusão de todo o processamento, será enviado para o cliente.

SGCPI

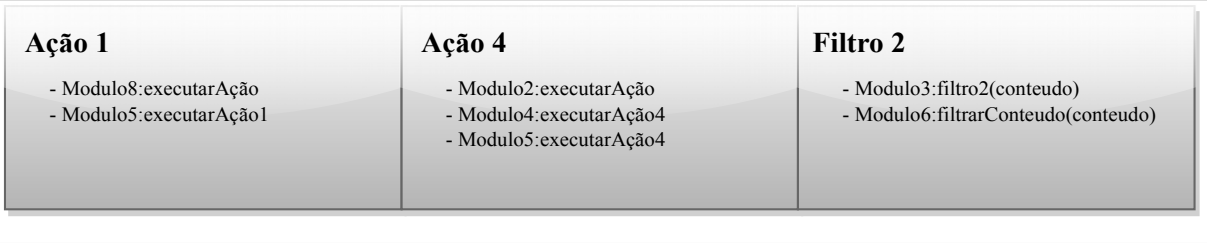


Figura 3.1: Representação da estrutura de dados para registo e controlo dos "ouvintes"

Na Figura 3.1 apresenta-se uma representação visual simplificada de uma possível estrutura de dados multidimensional onde para cada ação, filtro ou evento são associados os métodos ou funções a serem executadas no momento da invocação destas mesmas ações. A estrutura de dados pode ser complementada com outros elementos não representados na figura, tais como parâmetros adicionais para gestão de prioridades de execução dentro de um mesmo *hook*, registo dos métodos ou funções "ouvintes" já executadas, indicação de interdependências, etc.

3.2.2. Arquitetura proposta

Ao assumir o paradigma de programação orientado a eventos, podemos simplificar a arquitetura geral solução, através da implementação de um micro núcleo responsável por identificar e carregar cada um dos módulos específicos, fornecendo uma API simplificada para a gestão dos *hooks*.

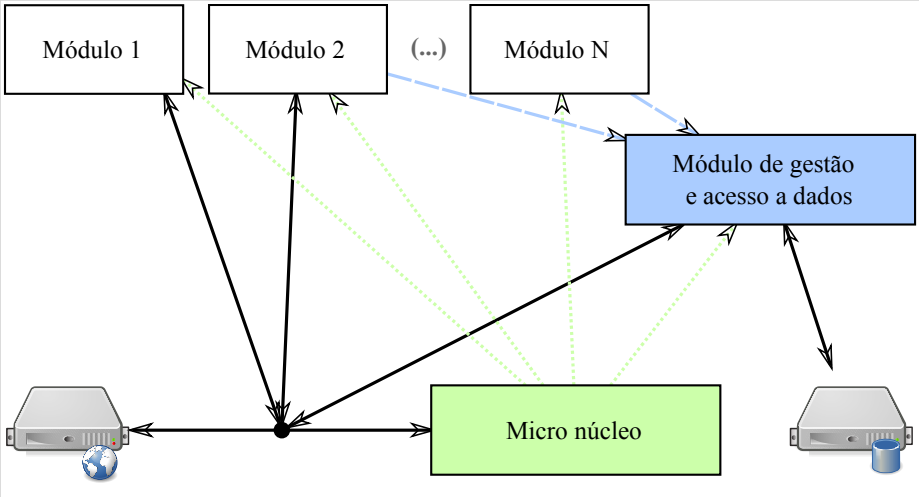


Figura 3.2: Arquitetura genérica da solução

Na Figura 3.2 encontra-se representada a arquitetura genérica da solução onde, numa primeira fase, o micro núcleo faz a indexação e carregamento dos módulos existentes, procedendo à sua inicialização, onde os módulos associam os seus métodos ou funções aos respetivos *hooks*. Após todos os módulos terem sido inicializados, o micro núcleo executa a ação principal à qual os módulos associaram os seus métodos com o comportamento e ações adicionais que pretendem ver executadas. Após todas as ações e respetivos métodos ou funções "ouvintes" terem sido executadas, o micro núcleo devolve o resultado final ao sistema, que devolve depois uma resposta ao cliente.

Este micro núcleo pode ser a base de toda a solução (Figura 3.2) ou suportar-se numa plataforma existente (Figura 3.3), sob a qual pode fornecer uma API intermédia para carregamento dos módulos e interligação da funcionalidade com a API fornecida pela plataforma escolhida.

SGCPI

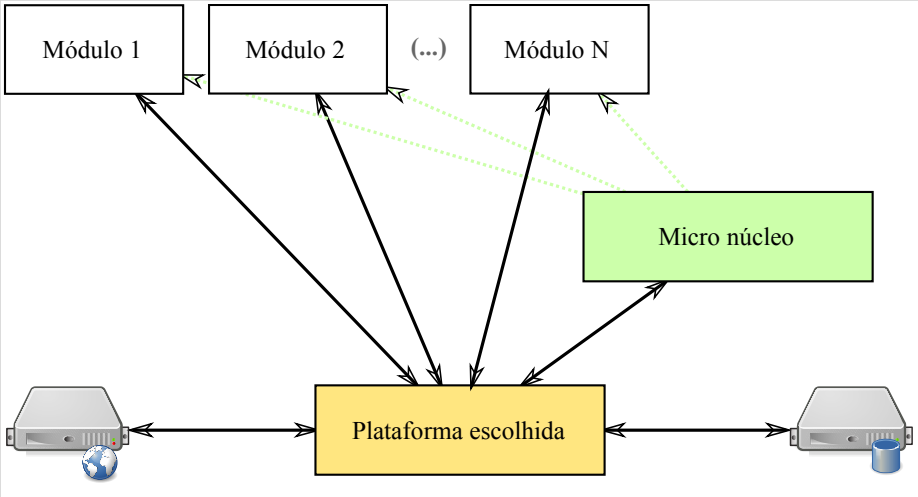


Figura 3.3: Arquitetura genérica da solução baseada numa plataforma existente

Neste cenário o micro núcleo indexa e inicializa cada um dos módulos, mas a gestão do fluxo de execução é assegurada pelo sistema base. Os módulos também podem comunicar diretamente com a API da plataforma base, modificação do seu comportamento em função dos *hooks* fornecidos.

4. Implementação da solução

Esta secção descreve a implementação da solução propriamente dita, nomeadamente questões como o processo de desenvolvimento de *software* utilizado (*Design-driven development*), apresentação da equipa base envolvida no projeto, incluindo as funções e responsabilidades do autor documentadas neste documento, plano geral do projeto, estudo comparativo das várias soluções analisadas e a plataforma base escolhida. Depois é apresentada a arquitetura final da solução baseada plataforma escolhida, as ferramentas, tecnologias e ambiente de desenvolvimento utilizado, estrutura dos temas e os vários módulos desenvolvidos com uma descrição do seu papel e funcionalidades, bem como os vários testes realizados à solução.

Dado o grande volume de módulos desenvolvidos para dar resposta a problemas muito específicos, esta secção apresenta uma grande quantidade de informação. Para facilitar a leitura do documento, alguma informação está disponível sob a forma de anexos devidamente referenciados ao longo da secção para que possam ser consultados para obtenção de informações e dados adicionais sobre o trabalho desenvolvido. Nesta secção destaca-se a importância de temáticas tais como a arquitetura final da solução (na página 37), a estrutura dos temas (que implementam a arquitetura descrita – página 49), os módulos específicos TinyMCE (para personalização do editor HTML da solução – página 58), ContentTemplates (para criação de modelos virtuais de conteúdo – página 65), PostCustomFields (para definição de campos de meta-dados – página 86) ou o módulo ResponsiveImages (para gestão de imagens responsivas – página 96). Destaca-se ainda sub-secção testes de desenvolvimento (na página 116) com a descrição de alguns dos testes de qualidade efetuadas à solução implementada.

4.1. Processo de desenvolvimento de *software*

A metodologia de desenvolvimento de projetos de *software* na UED não se enquadra completamente nas metodologias existentes, mas podemos dizer que incorpora algumas caraterísticas do *Extreme Programming* (XP) e do Scrum. Adota o processo de desenvolvimento *Design-driven development* (D3) [13] para identificação dos requisitos da solução onde, após uma reunião com o *sponsor*/cliente do projeto (*product owner* do Scrum), são desenvolvidas as propostas gráficas numa ferramenta de composição de imagem que representam o funcionamento geral da solução tal como foi descrito pelo *sponsor* e interpretado pela *designer* (exemplos no Anexo XI ao Anexo XXI). Estas propostas gráficas são desenvolvidas em estreita parceria entre o *designer* e o programador onde este último se pronuncia sobre a exequibilidade técnica da proposta gráfica em micro-reuniões de trabalho de poucos minutos, ao longo do processo de desenho das propostas.

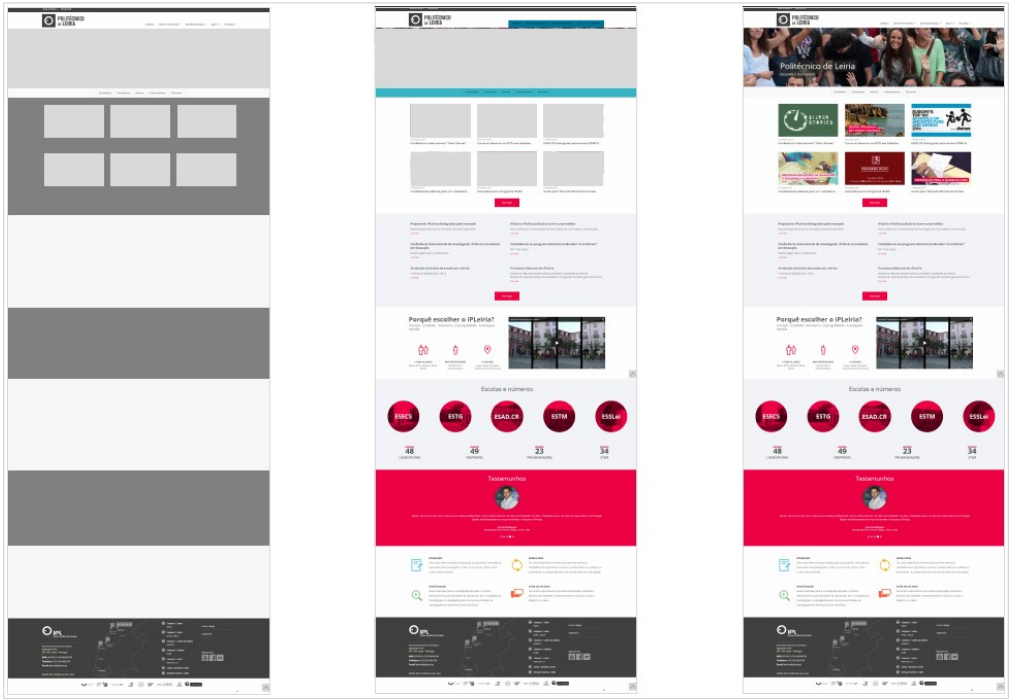


Figura 4.1: Etapas da construção de uma proposta gráfica

Quando a equipa está satisfeita com os resultados, estes são apresentados ao cliente que se pronuncia sobre os mesmos, sendo depois feitos os ajustes considerados necessários com base nas críticas recebidas. Após a aprovação das propostas gráficas, é iniciado o processo de implementação onde o sistema a ser desenvolvido deverá produzir resultados em tudo semelhantes aos representados na proposta gráfica.

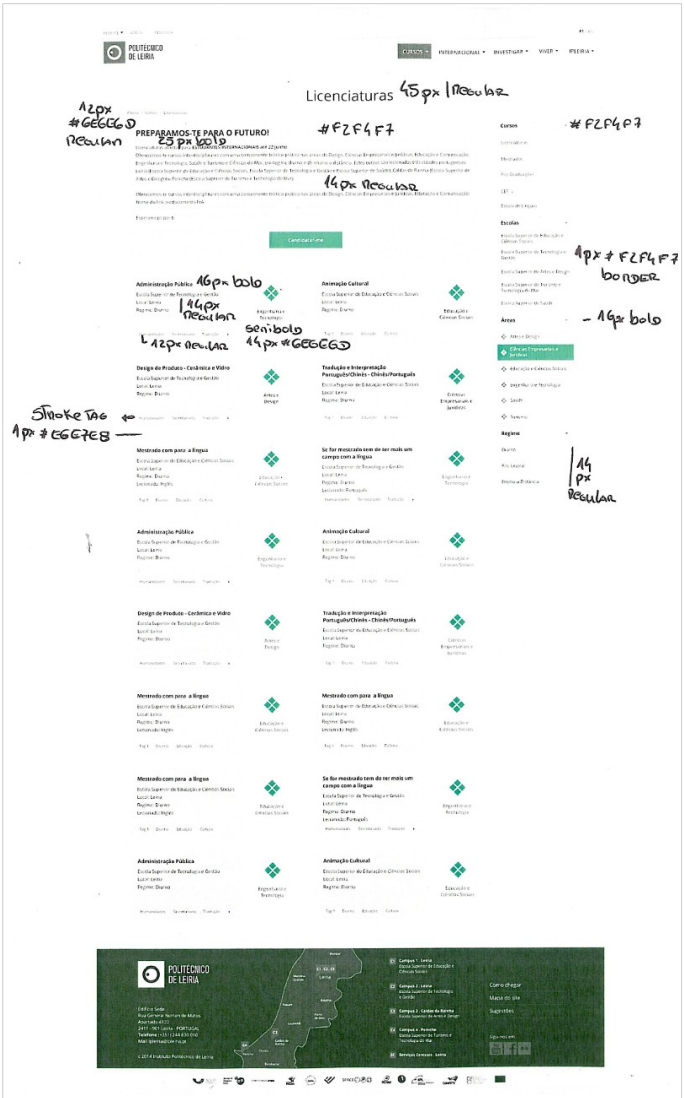


Figura 4.2: Informações técnicas para implementação de uma página

Na eventualidade do programador encontrar um problema não identificado na fase inicial de conceção da proposta (por exemplo, um problema de acessibilidade ou de incompatibilidade com determinada funcionalidade), esta questão é debatida entre os elementos da equipa no sentido de encontrar uma solução para o problema. No entanto verifica-se que quando este tipo de situações acontece, trata-se de problemas essencialmente funcionais que geralmente não influenciam drasticamente o aspeto gráfico proposto nem o compromisso assumido com o cliente, permitindo que estas questões sejam resolvidas com introdução de pequenas correções ou através da eliminação ou substituição dos pequenos artefactos gráficos que estejam na origem do problema.

SGCPI

Na Figura 4.2 encontra-se representado um exemplo de uma proposta gráfica, tal como foi fornecida ao programador, para implementação de um sistema para a listagem de cursos com aplicação de filtros. Com esta informação e o acompanhamento que o programador faz ao longo do processo de *design* da mesma, este tem a noção exata do aspeto final da solução e de como a mesma deve funcionar. Por outro lado, quaisquer recursos gráficos (como fotografias, ícones e ilustrações) que sejam necessários para o processo de desenvolvimento podem ser facilmente extraídos com as dimensões adequadas a partir do ficheiro fonte da proposta.

Com a adoção do processo D3 verifica-se que muitas vezes não existe a necessidade de se fazer o levantamento de requisitos por parte do programador, dado que este trabalho acaba por ser feito pelo *designer* para a criação das propostas gráficas que são, geralmente, auto-explicativas. No entanto, verifica-se que depois da proposta gráfica estar desenvolvida, torna-se mais simples fazer a identificação dos requisitos específicos que não foi possível inferir através da análise visual das propostas. Por exemplo, assumindo o exemplo da Figura 4.2, da análise visual não é possível identificar de forma imediata respostas a questões tais como quem pode gerir a informação, quais os tipos de filtros que podem ser aplicados, como é feita a gestão desses filtros, etc. Este tipo de questões são geralmente esclarecidas com o cliente, ou mesmo com o *designer*, que acaba por ter uma visão muito concreta da lógica de negócio da solução.

Verifica-se que para que este processo de desenvolvimento funcione, é necessário existir uma forte relação de trabalho colaborativo entre os vários elementos que compõem a equipa, onde os elementos se complementam, numa equipa verdadeiramente multidisciplinar. Este processo de desenvolvimento de *software* é utilizado à vários anos na maioria dos projetos desenvolvidos pela UED, embora nunca tenho sido aplicado em nenhum projeto desta dimensão.

4.2. Equipa de projeto

Apesar do trabalho descrito neste documento se centrar no trabalho desenvolvido pelo seu autor, este fazia parte de uma equipa multidisciplinar cujos membros assumem papéis muito específicos ao longo do ciclo de desenvolvimento da solução, como geralmente acontece em projetos desta natureza e dimensão. No âmbito deste projeto foi constituída a seguinte equipa de trabalho:

- Rita Cadima - *Sponsor*/*product owner* e gestora de conteúdos. Representou a presidência do IPLeiria no projeto e foi responsável pela seleção dos conteúdos a serem disponibilizados no portal.

- Joana Mineiro – *Lead designer* responsável por toda a identidade gráfica do projeto, nomeadamente estrutura e organização da informação, cores, iconografia, etc.

- Cláudio Esperança – *Lead developer*, *DevOps*, *front-end developer*, formador e administrador de sistema. Responsável pela arquitetura, desenho, implementação e administração de toda a solução tecnológica, bem como a formação dos gestores de conteúdos do *site*. Autor deste documento.

- Catarina Maximiano – *front-end developer*, formadora e administradora de sistema, responsável pela gestão de conteúdos, implementação de grande parte das propostas gráficas dos *sites* que compõem a solução. Implementação e especialização de alguns módulos a partir de módulos existentes, formação aos gestores de conteúdos e suporte técnico.

- Sandro Costa – *front-end developer*, formador e administrador de sistema, responsável pela gestão de conteúdos e implementação de algumas das propostas gráficas no portal principal da solução. Implementação de funcionalidades específicas sobre a solução,

formação aos gestores de conteúdos e suporte técnico.

- Vítor Rodrigues – administrador de sistema responsável pela gestão da infraestrutura e comunicação com a equipa de operações do IPLeiria.

- Nelson Matias e Paulo Gomes – elementos da equipa de operações do IPLeiria, responsáveis pela administração e instalação do sistema operativo e serviços base de apoio à solução. Responsáveis pela implementação da política de cópias de segurança bem pelas atualizações de segurança ao sistema operativo e respetivos serviços base.

Para destes elementos da equipa base, participaram no projeto dezenas de outras pessoas em tarefas como a gestão e validação de conteúdos, testes à solução, inserção de informação, entre outras. No total e na primeira fase do projeto estiveram envolvidas cerca de 50 pessoas para além dos elementos identificados na equipa base do projeto.

4.3. Plano de projeto

O plano do projeto foi composto pelas seguintes etapas:

- Reuniões com a direção, equipa de desenvolvimento e utilizadores da solução anterior para identificação das necessidades, problemas e análise do funcionamento da solução anterior;
- Análise da estrutura e informação disponibilizada pela solução anterior;
- Prototipagem gráfica;
- Identificação de requisitos genéricos;
- Instalação e configuração do ambiente de testes e de desenvolvimento;
- Teste e análise de soluções existentes e implementação de algumas *spikes* para testes de conceitos;
- Parametrização do ambiente de controlo de versões e definição do *workflow* de

desenvolvimento para o projeto;

- Implementação da arquitetura base da solução;
- Desenvolvimento dos módulos comuns;
- Concretização das propostas gráficas (com recurso ao processo de desenvolvimento D3);
- Testes e validações técnicas;
- Instalação da solução num ambiente de pré-produção;
- Importação e reorganização dos conteúdos identificados pelo *product owner* da solução anterior;
- Instalação e configuração de uma solução de análise estatística ao tráfego, de e para o sistema;
- Testes internos com utilizadores e gestores de conteúdos;
- Análise dos resultados dos testes e implementação das correções para os problemas identificados;
- Formação às equipas de gestão de conteúdos;
- Apoio à atualização dos conteúdos por parte das equipas de gestão de conteúdos;
- Configuração do ambiente de produção;
- Definição e implementação da política de cópias de segurança;
- Importação dos conteúdos do ambiente de pré-produção para o ambiente de produção e disponibilização da solução ao público em geral;
- Monitorização e acompanhamento da solução;
- Implementação de otimizações e melhoramentos;
- Documentação do trabalho desenvolvido.

Estas etapas dizem apenas respeito à 1ª fase do projeto onde se pretendia desenvolver uma solução que substituísse o portal anterior do IPLeiria, os portais de divulgação da oferta formativa e o *site* do Guia do Estudante do IPLeiria. Durante o ciclo de desenvolvimento foi

ainda solicitado pela presidência do instituto, que se implementasse também os *sites* dos Serviços de Documentação e da Direção dos Serviços de Informática sob este novo sistema. Todos os restantes portais institucionais seriam implementados numa 2ª fase, depois da análise dos resultados obtidos após a disponibilização desta solução.

4.4. Estudo de soluções

Como foi discutido na secção 2, existe uma grande quantidade de soluções para a gestão de conteúdos na Web, desde soluções *open-source* a soluções proprietárias, de soluções à medida a soluções mais genéricas. Tecnicamente é difícil afirmar que uma solução é melhor do que outra, mesmo porque esta afirmação depende do contexto, dos objetivos a que se destina, da equipa envolvida no projeto, da existência ou não de experiência anterior com a solução e/ou tecnologias que a suportam, flexibilidade do sistema, segurança, desempenho, entre muitos outros fatores.

No âmbito deste projeto foram analisadas algumas soluções que serão discutidas nas sub-secções seguintes.

4.4.1. Solução de raiz e à medida

Uma solução desenvolvida de raiz e à medida apresenta uma grande vantagem: existe um controlo absoluto sobre o sistema, o que permite adaptar o mesmo às necessidades mais específicas do projeto. No entanto, planear, desenhar, implementar, testar, otimizar, validar e distribuir cada funcionalidade, como partes integrantes de um projeto com alguma dimensão, requer uma grande quantidade de recursos, sejam estes temporais, humanos, materiais, organizacionais e económicos, que permitam garantir que os objetivos propostos são atingidos. Em soluções deste tipo não podem ser desprezadas as questões relacionadas com a segurança, desempenho e fiabilidade (sob pena da solução falhar no momento mais crítico), o

que obriga a testes e a um acompanhamento exaustivo que permita detetar e corrigir os problemas numa fase embrionária da planificação ou, no limite, na fase de desenvolvimento.

Esta foi uma das abordagens que o IPLeiria tentou no passado mas que acabou por ser abandonada. De qualquer modo, dada a urgência na disponibilização da solução (requisito RNF013) e os recursos disponíveis para o desenvolvimento da mesma, considerou-se que não existia tempo suficiente para desenvolver uma solução de raiz, em tempo útil, com todas a garantias de segurança, fiabilidade, desempenho e qualidade.

4.4.2. SharePoint

Esta foi uma plataforma base de alguns *sites* da instituição durante alguns anos, servindo de solução tecnológica a alguns portais do IPLeiria. Esta solução já havia sido analisada e testada anteriormente durante alguns meses para avaliar a sua aplicabilidade nos projetos desenvolvidos pela UED. Da análise dos resultados desta análise e das discussões com os gestores de conteúdo que trabalharam diretamente com a solução em uso nos portais do IPLeiria, concluiu-se que os problemas de usabilidade, acessibilidade e as limitações tecnológicas (por exemplo, para a criação em tempo útil de modelos gráficos com uma identidade própria em função dos projetos a que destinavam), eram demasiado evidentes para que esta solução pudesse ser novamente adotada neste projeto. Por outro lado, os custos adicionais com licenciamento e *hardware* necessários para garantir o correto funcionamento da solução eram também um forte entrave à escolha desta solução. Por fim, tendo a presidência do IPLeiria pedido uma alteração do paradigma funcional anterior, considerou-se que não faria sentido a adoção de uma solução que pudesse apresentar o mesmo tipo de problemas.

4.4.3. Joomla!, Drupal e WordPress

Estas são possivelmente as plataformas CMS mais utilizadas na Web atualmente. São soluções *open-source* onde, a disponibilização do seu código fonte, torna possível adaptar cada solução às necessidades mais específicas de cada projeto onde são utilizadas. Por outro lado, o acesso ao código fonte permite conhecer em detalhe os pormenores de implementação de cada uma das soluções, o que permite avaliar opções de arquitetura, preocupações de segurança e de desempenho, bem como facilitar o processo de depuração e resolução de problemas, comparativamente ao que acontece com uma solução fechada (modelo *black box*), sem acesso ao código fonte.

Joomla!

O Joomla! é uma solução com alguma expressão no mercado, principalmente em *sites* de comércio eletrónico ou redes sociais. É extensível com recurso às 8928 extensões disponíveis no seu diretório de extensões, das quais apenas 4405 são gratuitas. De facto, esta é uma das principais diferenças relativamente às restantes plataformas *open-source* analisadas, onde se confirma que a vertente comercial é muito mais proeminente no Joomla!. Suporta temas, embora a maioria tenham custos associados. Funcionalmente é uma solução com uma dificuldade de administração mediana, quando comparada com as outras plataformas analisadas, com imensas opções configuração (o que pode confundir alguns utilizadores). Da análise dos comentários da comunidade verificou-se que ocorrem problemas em processos de atualização, algumas extensões apresentam problemas graves de segurança que comprometem o funcionamento de toda a solução e existem também queixas de lentidão em *sites* com um volume significativo de conteúdos. Por fim, o suporte *multi-site* identificado como requisito para este projeto depende de extensões comerciais, cujo suporte apresenta custos relativamente elevados.

Drupal

O Drupal foi utilizado durante alguns anos como plataforma tecnológica de suporte ao *site* da UED. É uma solução robusta, segura e com um bom desempenho. Também é extensível com recurso a extensões, sendo que diretório oficial de extensões apresenta 16.692 módulos disponíveis, dos quais apenas 10.269 são compatíveis com a última versão estável da plataforma. Tem suporte a temas, estando 1.310 disponíveis no diretório de extensões da plataforma, sendo apenas 640 compatíveis com a última versão estável.

O facto da API sofrer por vezes alterações significativas entre versões principais, pode levar a que as personalizações efetuadas tenham de ser reimplementadas. Tecnologicamente é a solução mais complexa das plataformas analisadas e apresenta uma API com uma estrutura mais restritiva do que a do WordPress. Tem suporte nativo a *multi-site*, embora existam algumas preocupações de segurança relativamente à possibilidade de execução de código por parte de administradores locais, o que pode comprometer a segurança dos restantes *sites* disponibilizados na solução [14].

WordPress

Pela análise dos comentários da comunidade, o WordPress é uma das plataformas mais simples de utilizar, sendo dada especial atenção às questões de usabilidade durante os ciclos de desenvolvimento do produto. Tem um diretório oficial com 36.584 extensões e 1.714 temas gratuitos. A instalação de módulos pode ser feita diretamente no sistema de ficheiros do servidor ou através da área administrativa da plataforma. O suporte a *multi-site* é fornecido nativamente pelo sistema, embora a configuração desta funcionalidade careça de conhecimentos técnicos mais aprofundados.

Estima-se que o WordPress seja utilizado em 60% de todos os *sites* que utilizam um CMS [8]. O facto de ter uma enorme base instalada faz com que esta solução tenha uma das mais

maiores e mais ativas comunidades a suportar o projeto, o que garante alguma segurança e rapidez na resolução dos problemas. Uma das linhas orientadoras do desenvolvimento do WordPress é a de nunca quebrar a retro-compatibilidade com versões anteriores do WordPress, facilitando do processo de atualizações para os utilizadores e programadores [8]. Nas versões mais recentes foi ativado um mecanismo de atualizações automáticas para versões menores, garantindo que as atualizações de segurança são aplicadas na plataforma sem intervenção dos administradores do sistema. Por fim, e ainda relativamente à questão de segurança, o WordPress conta com uma equipa de segurança própria, com aproximadamente 25 pessoas [8], entre programadores e especialistas de segurança, para além de muitos outros elementos, que fazem parte da comunidade que suporta a solução. Esta equipa monitoriza continuamente a plataforma, respetivas extensões e temas, em busca de vulnerabilidades e de problemas de segurança.

Tecnicamente a documentação é bastante completa e o código fonte está bem documentado. A API não é tão estruturada ou rígida como nas outras plataformas, o que permite alguma flexibilidade no desenvolvimento de soluções à medida. O WordPress faz uso do paradigma de programação orientada a eventos, através das ações e dos filtros [15], permitindo que, em momentos concretos do fluxo de execução associado ao processamento de um determinado pedido, sejam executadas as funções ou métodos necessários para modificar o comportamento da solução. Por exemplo, um módulo pode registar uma função para ser executada quando a função `wp_head` do WordPress é invocada no cabeçalho da página, permitindo injetar código adicional nessa secção. Outro exemplo seria a aplicação de um filtro através da associação de uma função de um módulo ao filtro `the_content`, o que permitiria que essa função processasse um conteúdo e devolvesse um resultado que seria utilizado como o conteúdo com uma funcionalidade adicional na página a ser apresentada. A API que permite implementar esta funcionalidade está também disponível para os módulos que não façam parte da instalação base do WordPress, permitindo que estes troquem informações e executem ações próprias entre si. No caso de uma determinada ação não existir ou não ser executada num determinado fluxo de execução associado ao processamento de um pedido específico, as

funções associadas a essa ação são simplesmente ignoradas garantindo que o sistema continua a funcionar sem comprometer a sua estabilidade. Com este sistema é possível criar uma arquitetura modular, estendendo e adaptando as funcionalidades do WordPress às necessidades específicas de cada projeto [16].

4.4.4. Conclusão

Após a análise das vantagens e desvantagens de cada uma das soluções, nomeadamente questões como a filosofia de desenvolvimento de *software*, a extensibilidade da API, a adequação aos requisitos e a experiência anterior com solução, entre outros fatores, a escolha do sistema base recaiu sob a plataforma WordPress, sob a qual seria desenvolvida uma arquitetura modular que complementasse ou modificasse o funcionamento do sistema adaptando-o às necessidades específicas deste projeto.

4.5. Arquitetura final da solução

O WordPress começou por ser um sistema CMS muito direcionado para criação e gestão de blogues. Atualmente, apesar de continuar a ser muito utilizado para este fim, assume-se claramente como uma das plataformas CMS genéricas de referência. Um dos principais motivos que fundamenta esta escolha é a sua extensibilidade onde, através de extensões ou de temas, podemos adicionar ou personalizar funcionalidades da ferramenta de acordo com as preferências ou necessidades dos projetos desenvolvidos.

Na arquitetura final da Figura 4.3, construída a partir da adaptação da arquitetura geral proposta na secção 3.2.2, temos uma solução *multi-site* WordPress onde cada projeto (com uma identidade ou funcionalidades específicas) terá o seu próprio tema, associado ao *site* específico desse projeto, numa única instalação partilhada do WordPress. Isto permite otimizar recursos tecnológicos, centralizando a solução num único sistema, facilitando o

processo de gestão da plataforma, criação de *sites*, autenticação, gestão de utilizadores, atualizações de segurança, cópias de segurança, etc. Para os utilizadores a solução é totalmente transparente, dado a sensação que cada *site* é uma instalação própria isolada, quando na realidade se trata de um ambiente partilhado.

As aspetos funcionais principais de cada *site* estão encapsulados em módulos associados ao tema específico de cada *site*, fornecendo todas as funcionalidades necessárias para o seu correto funcionamento, de forma completamente isolada aos restantes *sites* do sistema.

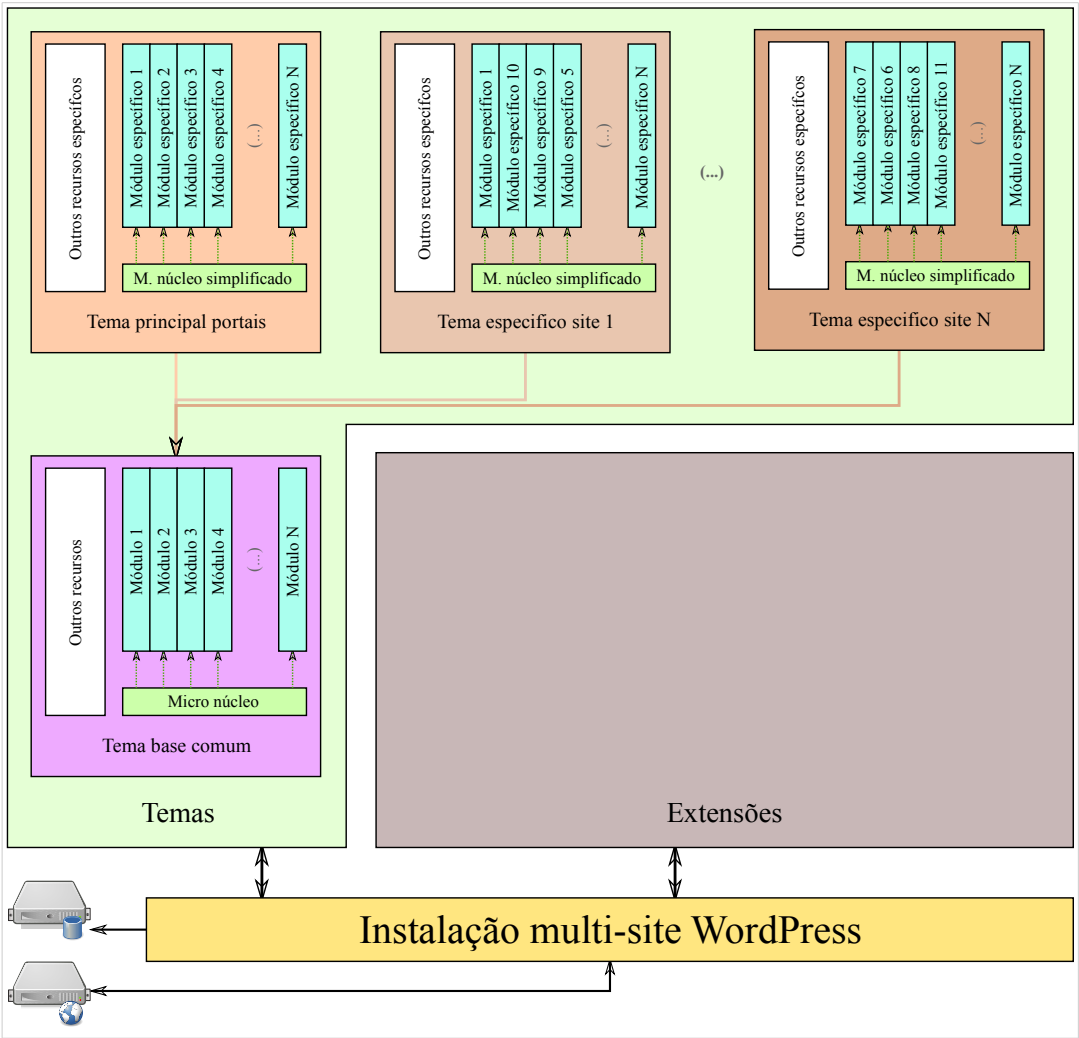


Figura 4.3: Arquitetura final da solução

Nesta arquitetura temos uma instalação *multi-site* onde associado a determinado endereço temos um *site* próprio com um tema específico, com a possibilidade de ter ativas ou não extensões adicionais que estejam disponíveis no sistema. Quando o sistema carrega o tema associado ao *site*, o micro-núcleo simplificado do mesmo indexa e carrega os módulos próprios associados a esse tema e carrega o micro-núcleo do tema base, comum a todos os temas. Por sua vez, este micro-núcleo do tema base indexa e carrega também os módulos comuns que controla. O processo de inicialização de cada módulo fica ao cargo dos mesmos, pertencendo a estes a responsabilidade da associação dos seus métodos aos *hooks* fornecidos pela API do WordPress. Em caso de necessidade os módulos dos temas podem ainda comunicar entre si, bem como com a API das extensões que estejam ativas para esse *site* no sistema. No entanto cada módulo deve garantir que a não-existência de um determinado módulo ou extensão da qual a sua funcionalidade dependa não compromete o normal funcionamento da plataforma.

Apesar de concecionalmente as extensões serem os mecanismos privilegiados para estender funcionalidades no WordPress, com exceção de algumas funcionalidades específicas associadas à gestão de extensões, não existem tecnicamente grandes limitações que impeçam o encapsulamento de aspetos funcionais relacionados com a API do WordPress em temas. No âmbito desta solução consideraram-se as vantagens desta abordagem, permitindo incorporar toda a funcionalidade necessária para o correto funcionamento de um *site* no tema que o suporta. Isto permite que, para módulos comuns utilizados em vários *sites*, cada *site*/tema personalize de forma independente o funcionamento desses módulos em função da proposta gráfica e das necessidades específicas desse sub-projeto, sem preocupações de comprometer a funcionalidade para os restantes *sites* que utilizem um mesmo módulo. Esta liberdade contrapõem-se ao cenário onde seriam utilizadas extensões genéricas comuns a vários *sites*, onde a personalização ou evolução de determinada extensão poderia comprometer o funcionamento do sistema em outros projetos que dela dependam.

O WordPress fornece ainda uma funcionalidade de herança de temas, onde um determinado

tema pode estender as funcionalidades de um tema pai, implementando as suas próprias funcionalidades e personalizações. A arquitetura proposta tira partido desta funcionalidade para englobar num tema pai/base todas as funcionalidades/módulos comuns aos vários temas e que serão herdadas por estes e onde não exista a necessidade de personalização específica desses módulos em função dos projetos onde será utilizados. Por exemplo, um módulo para implementação de técnicas de SEO será um módulo que fará sentido ser comum a todos os projetos e onde não existe necessidade de reimplementação da sua funcionalidade nos sub-projetos desenvolvidos. Esta abordagem permite reduzir consideravelmente o tamanho dos temas específicos de cada *site*, implementando apenas as funcionalidades adicionais ou alterações gráficas específicas dos projetos aos quais estão associados.

4.6. Ferramentas de desenvolvimento

4.6.1. Ambiente de desenvolvimento

O ambiente de desenvolvimento foi baseado na pilha de *software* LAMP, ou seja, um sistema operativo GNU/Linux, o servidor Web Apache, o SGBD MySQL e o interpretador de linguagem PHP. Os principais motivos para a escolha deste conjunto de *software* prendem-se com a excelente compatibilidade com a plataforma WordPress, estabilidade da solução, desempenho, proximidade com o ambiente de produção e conhecimentos anteriores da tecnologia.

Para o desenvolvimento da solução propriamente dita foi utilizado o IDE NetBeans com as extensões para Git, CSS, PHP e HTML5. Esta ferramenta permitiu acelerar consideravelmente o desenvolvimento através da indexação do código-fonte de toda a solução, permitindo uma consulta quase imediata da implementação de determinadas funcionalidades na API do WordPress. Para edição de *scripts* SQL de grandes dimensões foi utilizado o VIM como editor de texto.

Para implementação de uma réplica do ambiente de produção para execução de testes específicos, foi utilizado o Vagrant[6] para configuração de um ambiente virtual de desenvolvimento, em conjunto com o *software* de virtualização VirtualBox[7] e o Puppet[8] (sistema de gestão de configurações). Ao ser iniciado, o Vagrant configurava ou inicializava uma máquina virtual com toda a solução, com uma base de dados completa e com os ficheiros de conteúdos obtidos a partir do servidor de produção. O resultado era uma cópia muito fiel do sistema em produção, no qual foram efetuados diversos testes e validadas configurações, antes das mesmas serem implementadas no servidor de produção.

4.6.2. Controlo de versões

Dada a dimensão do projeto e os vários programadores envolvidos no desenvolvimento da solução, era impensável não recorrer a um sistema de controlo de versões para acompanhar o trabalho desenvolvido, manter um histórico de alterações, gerir a integração de código, facilitar o desenvolvimento de novas funcionalidades e a correção de problemas no sistema, etc. Para este projeto escolheu-se o Git[9] como VCS não só pelo seu desempenho, comprovado em projetos anteriores, como por não forçar um paradigma de controlo de versões centralizado, permitindo o desenvolvimento de código de forma paralela e distribuída através da utilização de repositórios locais geridos por cada programador.

Para facilitar e centralizar o processo de gestão de projetos de *software*, o IPLeiria adotou a plataforma *open-source* Redmine[10], uma solução que permite não só planificar, documentar e acompanhar o desenvolvimento de projetos deste tipo, como fornece suporte direto a várias ferramentas de controlo de versões, incluindo o Git.

6 https://www.vagrantup.com/

7 https://www.virtualbox.org/

8 https://puppetlabs.com/

9 http://git-scm.com/

10 http://www.redmine.org/

SGCPI

Últimas revisões

#			Data	Autor	Comentário
134e993e	◉		17/02/2015 19:04	Cláudio Esperança	Issue with post year resolved on the Agreements module;
8e102e16	○	◉	16/02/2015 20:35	Cláudio Esperança	Grunt fix to concat CSS on a single CSS theme file to work correctly with apache pagespeed module;
eccd711e	○	○	16/02/2015 17:18	Cláudio Esperança	Merge branch 'master' of ssh://redmine.ipleiria.pt/p0061/p0118/p0161/wordpress-theme-ipleiria_ipleiria
b035e844	○	○	16/02/2015 15:09	Catarina Maximiano	Merge branch 'master' of ssh://redmine.ipleiria.pt/p0061/p0118/p0161/wordpress-theme-ipleiria_ipleiria
a5a17e38	○	○	16/02/2015 15:08	Catarina Maximiano	add agreements content type
94296b10	○	○	14/02/2015 04:13	Cláudio Esperança	Language pack update;
86c7e999	○	○	13/02/2015 20:03	Cláudio Esperança	New IPLeiria Top Bar and Footer systems;
13b72631	○	○	09/02/2015 15:32	Sandro Costa	Final adjustments to the news stylesheet to have the line-height number a bit lower
0d6e7b8c	○	○	06/02/2015 17:59	Sandro Costa	minor changes to the post-type news template to feature accessible links and better spacing between text elements
ec7bcca3		○	06/02/2015 16:54	Sandro Costa	Changed posttype template news to avoid an issue with a double line excerpt becoming a single line.

Figura 4.4: Exemplo do histórico das últimas revisões de um projeto

Isto permitiu que se criasse vários repositórios para vários projetos, facilitando o desenvolvimento de trabalho de forma paralela e distribuída, sob os quais qualquer programador autorizado podia integrar alterações com novas funcionalidades ou melhoria de funcionalidades existentes ao longo do ciclo de vida do projeto.

4.6.3. Ferramentas adicionais de desenvolvimento

Apesar do paradigma de desenvolvimento de *software* de projetos Web ainda se centrar muito em linguagens interpretadas, nos últimos tempos têm-se assistido à introdução de algumas técnicas e tecnologias inspiradas no paradigma das linguagens compiladas, que otimizam o código-fonte de modo a aumentar ao máximo o desempenho dos projetos aos quais

disponibilizam funcionalidade. Minificação, ofuscação e reorganização de código, *responsive design* e *graceful degradation*, são apenas algumas das técnicas utilizadas atualmente no desenvolvimento de projetos Web. No âmbito deste projeto foram estudadas e implementadas algumas destas técnicas no sentido de adotar as boas práticas vigentes para a temática em causa.

Meta-informação

Todos os temas desenvolvidos foram acompanhados de um ficheiros `package.json` com alguns meta-dados do projeto, tais nome, versão, descrição, autor, licença, informação sobre o repositório, dependências, etc.

```
{
  "engines": {
    "node": ">= 0.10.0"
  },
  "devDependencies": {
    "grunt": "~0.4.2",
    "grunt-contrib-jshint": "~0.7.2",
    "grunt-contrib-watch": "~0.5.3",
    "grunt-contrib-compass": "~0.5.0",
    "grunt-contrib-cssmin": "~0.6.2",
    "grunt-contrib-uglify": "~0.2.7",
    "grunt-concurrent": "~0.5.0",
    "grunt-autoprefixer": "~0.3.1",
    "grunt-csso": "~0.6.3",
    "grunt-contrib-concat": "~0.5.0"

  },
  "name": "wordpress-theme-ipleiria",
  "version": "0.6.0",
  "description": "WordPress theme for the IPLeiria website",
  "homepage": "http://ipleiria.pt",
  "repository": {
    "type": "git",
    "url": "ssh://git@redmine.ipleiria.pt/p0061/p0118/wordpress-theme-ipleiria.git"
  },
  "keywords": [
    "IPLeiria",
    "WordPress",
    "Theme"
  ],
  "author": "Cláudio Esperança <claudio.esperanca@ipleiria.pt>",
  "license": "GPLv3",
  "readmeFilename": "README.md"
}
```

Figura 4.5: Exemplo de um ficheiro `package.json` de um tema

SGCPI

Este ficheiro é utilizado por algumas das ferramentas complementares do projeto que utilizam estas informações para a sua parametrização. Por exemplo, com recurso ao comando `npm install` o gestor de pacotes do Node.js[11] (NPM[12]) recorre a este ficheiro para identificar as dependências do projeto e iniciar o processo de transferência, instalação e configuração dessas dependências.

Compass e SASS

O SASS[13] é uma extensão à linguagem CSS3 que adiciona suporte a regras encadeadas, variáveis, *mixins*, herança de seletores, entre outras funcionalidades. Permite gerar CSS bem formatado e torna as folhas de estilo mais simples de organizar e de manter [17]. No âmbito deste projeto, esta tecnologia foi utilizada em conjunto com a *framework* CSS Compass[14] para transformar as regras de estilo descritas no formato SASS para código CSS3 válido para ser interpretado pelos navegadores dos clientes dos vários *sites*. Isto permite que instruções repetitivas como definições de códigos de cor, tipos de letra, tamanhos de letra, etc., sejam centralizadas num único ficheiro onde estas variáveis são definidas para que possam depois ser utilizadas ao longo dos vários ficheiros SASS com as instruções CSS para a solução. Com esta técnica, a alteração de uma qualquer definição associada a uma variável ao longo do projeto está centralizada num único ponto, eliminando assim a necessidade do programador ter de inspecionar e interpretar centenas de instruções CSS em busca dos valores que seja necessário modificar.

CSS *Autoprefixer*

As tecnologias Web estão em constante evolução, introduzindo novas funcionalidades a um

11 https://nodejs.org/

12 https://www.npmjs.com/

13 http://sass-lang.com/

14 http://compass-style.org/

ritmo alucinante, fazendo com que a definição de normas para a Web geralmente aconteça depois das tecnologias terem sido amplamente testadas e validadas no mercado. No caso dos novos estilos CSS utilizados na Web, a forma como a maioria dos produtores de navegadores introduz estas novidades é através de prefixos do motor de renderização (*vendor prefixes*), que podem ser utilizados pelos programadores de projetos Web para testar e validar a implementação, enquanto a tecnologia é normalizada através de um *standard* comum. Por exemplo, a possibilidade de utilização da propriedade `box-shadow` (que permite adicionar uma sombra no formato caixa a um elemento HTML) começou por ser introduzida no motor de renderização WebKit através da propriedade `-webkit-box-shadow`. Quando esta definição passa a norma, o prefixo do motor desaparece (`-webkit-`), passando a funcionalidade a ser suportada oficialmente pelo navegador com o valor final da propriedade (`box-shadow`).

O CSS *Autoprefixer*[15] é uma ferramenta que analisa as instruções CSS em busca de propriedades não normalizadas que estejam definidas no código, adicionando ou removendo prefixos dos vários motores de renderização utilizados pelos navegadores, tornando o código mais compatível com os vários clientes existentes no mercado. Neste projeto a ferramenta foi parametrizada para suportar apenas as duas últimas versões dos navegadores mais utilizados, permitindo que a cada lançamento as instruções CSS evoluam consoante a evolução tecnológica introduzida pelos navegadores e processos de normalização.

CSS *Optimizer*

O CSS *Optimizer* (CSSO[16]) é uma ferramenta de redução e de otimização de código CSS. Para além da aplicação das técnicas habituais de redução de código (como a remoção de carateres de formatação, espaços duplicados, etc.), consegue também executar otimizações estruturais aos ficheiros CSS, reduzindo o tamanho dos ficheiros [18] e o *overhead* associado à transferência e interpretação do código por parte dos navegadores dos clientes.

15 https://github.com/postcss/autoprefixer

16 https://github.com/css/csso

SGCPI

Tecnicamente pode executar as seguintes transformações:

- Eliminação de espaços em branco;
- Eliminação o caráter de terminação `;`;
- Eliminação de comentários;
- Eliminação de declarações `@charset` e `@import` inválidas;
- Redução do tamanho da definição de propriedades de cor (por exemplo, transformação do valor `#FFFFFF` para `#FFF`);
- Eliminação de unidades de medida associadas ao valor `0`;
- Redução do tamanho associado a múltiplas linhas de instruções;
- Redução do tamanho associado à definição da propriedade `font-weight`.

Opcionalmente pode ainda executar as otimizações estruturais:

- Agregação de blocos com seletores e propriedades idênticas;
- Eliminação de propriedades (nos formatos expandido ou curto);
- Eliminação de seletores repetidos;
- Agregação ou separação parcial de blocos de instruções;
- Redução do tamanho associado à definição de propriedades `margin` e `padding`.

CSS*min*

O *CSSmin*[17] é uma ferramenta de redução de código CSS que, neste projeto, foi utilizada para reduzir o tamanho dos ficheiros CSS associados a código de terceiros (e cuja otimização com CSS *Optimizer* poderia introduzir problemas na reorganização das instruções).

17 https://www.npmjs.com/package/grunt-contrib-cssmin

Concat

O *Concat*[18] é uma ferramenta que permite agrupar vários ficheiros de acordo com regras definidas, permitindo assim reduzir o *overhead* associado à transferência de múltiplos ficheiros distintos.

JSHint

O JSHint[19] é uma ferramenta desenvolvida pela comunidade para detetar erros e potenciais problemas em código JavaScript, bem como notificar o programador para a violação de convenções de desenvolvimento adotadas [19].

Neste projeto, para além boas práticas definidas por omissão para a ferramenta, foram definidas algumas regras adicionais:

- Obrigatoriedade de aplicação de chavetas em blocos de código ou expressões condicionais;
- Aviso quando existem variáveis definidas mas não utilizadas;
- Supressão de avisos associados a comparações com valores nulos;
- Obrigatoriedade de definir uma variável antes de a utilizar;
- Proibição de utilização das instruções `arguments.caller` e `arguments.callee`;
- Proibição de utilização de variáveis não declaradas;
- Supressão de avisos associados à atribuição de variáveis em instruções de comparação;
- Supressão de avisos associados à utilização de variáveis definidas pelos navegadores, bem como pelo jQuery.

18 https://www.npmjs.com/package/grunt-contrib-concat

19 http://jshint.com/

UglifyJS

O UglifyJS[20] é uma ferramenta JavaScript para reduzir o tamanho de ficheiros com instruções JavaScript. Para além da remoção de espaços em branco duplicados e outros carateres não imprimíveis, pode aplicar algumas otimizações de código, tais como a remoção de variáveis e de código não utilizado, otimização de expressões condicionais, alterações de nomes de variáveis e funções locais para letras simples, etc. Com este módulo torna-se assim possível reduzir consideravelmente o tamanho dos ficheiros JavaScript reduzindo assim o tempo associado às transferências e à interpretação dos mesmos.

Grunt

Durante o ciclo de desenvolvimento de um projeto Web com alguma dimensão, existem várias tarefas que terão de ser repetidas centenas de vezes; é neste tipo de tarefas que o Grunt pode dar um contributo bastante relevante, na implementação de automatismos associados ao desenvolvimento de *software*. O Grunt[21] é uma ferramenta para automatização e execução de tarefas repetitivas, tais como redução de código, compilação, testes unitários, validação de código, entre outras. A configuração das tarefas é feita com recurso a um simples ficheiro (`Gruntfile.js`, `Anexo XXIII`) onde são dadas todas as indicações sobre as tarefas a executar e respetivas definições a utilizar. No âmbito deste projeto utilizou-se esta ferramenta para executar as ferramentas descritas anteriormente nesta secção, assumindo dois cenários distintos de execução: o cenário de desenvolvimento e o cenário de produção.

No cenário de desenvolvimento (comando `grunt watch` na raiz do tema), são executadas as tarefas associadas ao desenvolvimento, tais como a compilação dos ficheiros SASS, validação de código JavaScript e atualização dos ficheiros temporários a enviar para o navegador Web do cliente. Para automatizar este processo foi utilizado um módulo adicional responsável por

20 http://lisperator.net/uglifyjs/

21 http://gruntjs.com/

monitorizar as alterações a ficheiros de código fonte SASS ou JavaScript. Assim, sempre que um programador guardasse as alterações num ficheiro, este módulo detetava essas alterações e executava as tarefas associadas a esse tipo de ficheiro, disponibilizando de forma quase imediata as versões modificadas dos ficheiros necessários para apresentar as alterações efetuadas no cliente utilizado. Neste cenário não eram executadas quaisquer tarefas de otimização ou de redução de código em ficheiros, dado que o objetivo era que as alterações nos ficheiros temporários fossem apresentadas o mais rapidamente possível ao programador.

No cenário de produção (comando `grunt build` na raiz do tema), eram executadas todas as tarefas associadas à colocação do projeto em produção. Nesta etapa, para além de todas as tarefas executadas no cenário de desenvolvimento, são também executadas as tarefas de otimização e de redução de código em todos os ficheiros que tenham sofrido alterações desde a última execução do comando de produção.

Com as ferramentas descritas nesta secção foi possível aplicar uma série de otimizações e melhorias ao desempenho da solução onde, com recurso aos automatismos implementados, se reduziu o impacto da inclusão destas técnicas no tempo de desenvolvimento do projeto.

4.7. Estrutura dos temas

Dado que a maioria das funcionalidades específicas dos *sites* deste projeto está concentrada nos temas, importa descrever a sua estrutura lógica e física, para se ter uma ideia mais concreta do funcionamento do sistema tal como foi implementado.

Os temas do WordPress são geralmente compostos por quatro tipos de ficheiros:

- O **ficheiro de estilos**, um ficheiro obrigatório denominado por `style.css`, que contém os meta-dados do tema sob a forma de comentário com uma estrutura própria que apresenta informações como o nome do tema, versão, autor, etc. Opcionalmente (mas

preferencialmente) este ficheiro pode conter os estilos CSS que controlam a aparência geral do tema.

- Os vários **ficheiros PHP do modelo** que controlam o modo como as páginas do *site* geram a informação, a partir das estruturas de dados do WordPress, que são depois apresentadas ao cliente sob o formato HTML. Como exemplo temos o ficheiro obrigatório `index.php` utilizado para, com recurso à API do WordPress, processar o pedido e gerar o HTML a devolver ao cliente. Este é o ficheiro que é utilizado por omissão pelo WordPress se não for encontrado nenhum outro ficheiro mais adequado, dos definidos na hierarquia dos modelos do sistema (Anexo XXIV).

- A **biblioteca de funções PHP** específicas do tema, geralmente disponível no ficheiro opcional `functions.php`.

- **Outros recursos** opcionais, tais como imagens, ficheiros JavaScript, tipos de letra, etc., que podem ser utilizados para fornecer ou complementar o aspeto gráfico da solução.

A estrutura dos temas desenvolvidos no âmbito deste projeto evolui a partir deste estrutura. No entanto importa fazer a distinção entre a estrutura do tema base do sistema e os temas específicos de cada projeto que se baseiam no tema base.

4.7.1. Estrutura do tema base

O tema base fornece a funcionalidade que é comum a todos os temas específicos, permitindo uma partilha de funcionalidade entre os vários *sites* através dos módulos que são comuns a todos os sub-projetos.

A estrutura de ficheiros apresentada do tema `ipleiria` é apresentada na Figura 4.6 onde, na pasta `css` temos os ficheiros CSS principais do tema, gerados a partir da compilação dos

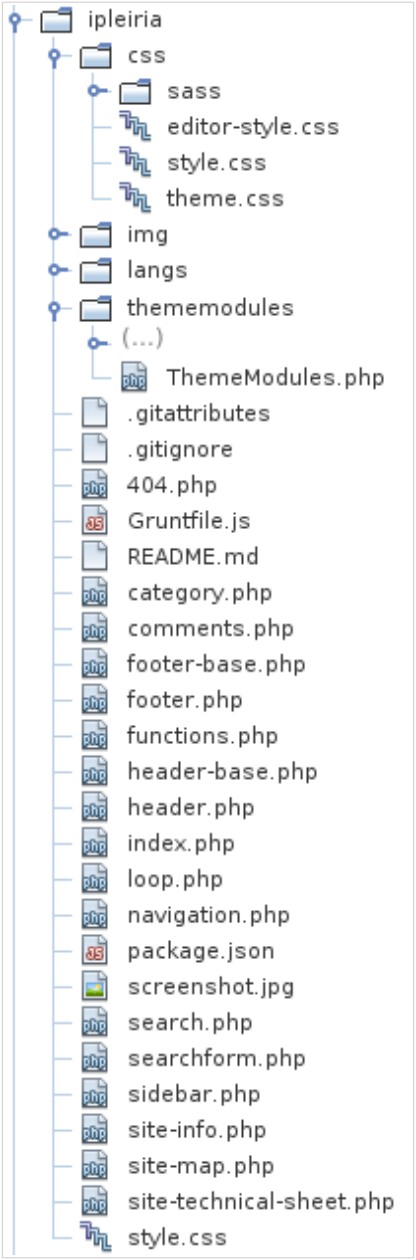


Figura 4.6: Estrutura de ficheiros do tema base

ficheiros SASS disponíveis na pasta `sass`, tal como foi descrito na secção 4.6.3. No momento da compilação os ficheiros intermédios `style.css` e `theme.css` gerados no passo anterior são agrupados num único ficheiro `style.css` com os estilos principais globais e que é colocado na raiz do tema para ser utilizado pelo motor de temas do WordPress. O ficheiro `editor-style.css` é um ficheiro especial utilizado para aplicar os estilos de conteúdo específicos ao tema, no editor WYSIWYG da plataforma, utilizado em algumas secções da área administrativa. Isto permite que o gestor de conteúdos apresente uma pré-visualização mais aproximada da aparência do conteúdo após este ser publicado durante a edição. A pasta `img` concentra, como o próprio nome indica, os recursos gráficos do tema em formato de imagem.

Os temas desenvolvidos no âmbito deste projeto fazem uso pleno da API de localização do WordPress para fornecer suporte a um interface em vários idiomas (na primeira fase, Português Ibérico e o Inglês, sendo este último o idioma nativo da solução). Assim, na pasta `langs` podemos encontrar os pacotes de idioma com os textos utilizados neste tema e que são apresentados contextualmente ao utilizador em várias secções dos *sites*.

A tradução pode ser feita com qualquer aplicação com suporte ao sistema de internacionalização e de localização

gettext, sendo que neste projeto a ferramenta adotada foi o PoEdit[22]. Por exemplo, para efetuar a tradução do tema para Português basta abrir o ficheiro `pt_PT.po` com os textos existentes e iniciar o processo de tradução de cada uma das frases. O catálogo foi parametrizado para suportar a atualização dos textos do catálogo a partir do código-fonte, através da configuração das palavras-chave utilizadas na API de localização do WordPress, pelo que textos que estejam associados a funções como `__`, `_e`, `esc_attr_e`, `esc_attr_e`, `esc_html__`, `esc_html_e` da API do WordPress são automaticamente importados para o catálogo sempre que o tradutor solicita uma atualização de textos a partir do código-fonte (Anexo XXV).

O ficheiro `.gitattributes` (Anexo XXVI) contém alguns atributos para parametrizar as estratégias de fusão (*merge*) para o sistema de controlo de versões Git, indicando quais são os ficheiros que devem ser mantidos quando se agregam dois repositórios de temas distintos (por exemplo, importa manter os ficheiros de meta-dados com as informações do tema, pacote de idioma, captura de ecrã, etc.). O ficheiro `.gitignore` informa o VCS quais os ficheiros que devem ser ignorados sempre que são feitas submissões de código para o sistema de controlo de versões (Anexo XXVII).

Como já foi referido na secção 4.6.3, a configuração das tarefas de desenvolvimento é feita através do ficheiro `Gruntfile.js`. A execução do comando `grunt` na raiz do tema com o parâmetro `watch` ou `build` é que vai determinar quais as tarefas que devem ser executadas, em que ordem e que parâmetros de configuração devem ser enviados a cada uma das ferramentas de desenvolvimento. Na raiz do tema temos também o ficheiro `package.json` com informações de meta-dados do projeto e que foi também referido anteriormente.

O ficheiro `README.md` é apenas um ficheiro informativo com alguma informação sobre o projeto (Anexo XXVIII). Este ficheiro é lido pelo Redmine e apresenta um documento estruturado com indicações fornecidas pelos programadores do sistema no repositório (Anexo XXIX). No contexto deste projeto, este ficheiro foi utilizado para partilhar informações com

22 http://poedit.net/

outros programadores, tais como comandos disponíveis, instruções de desenvolvimento, módulos desenvolvidos, estrutura do tema, entre outras.

A pasta `thememodules` contém o micro-núcleo (apresentado na Figura 4.3 da página 38), bem como todos os módulos do tema, disponibilizados como sub-diretórios nesta pasta. O ficheiro `ThemeModules.php` (Anexo XXX) contém a implementação em PHP do micro-núcleo, que consiste em:

- Um *trait* `tThemeModule` com a definição de algumas variáveis específicas e que os módulos podem utilizar para prevenir situações de uma dupla inicialização.

- Uma classe abstrata `ThemeModule` na qual todos os ficheiros base dos módulos se devem basear e que contém vários métodos auxiliares para fornecimento de funcionalidades comuns, tais como obtenção do endereço base do módulo (para carregamento de recursos adicionais, como folhas de estilo, ficheiros JavaScript, etc.), armazenamento e obtenção de valores associados a conteúdos na base de dados do WordPress, etc. No Anexo XXXI apresenta-se um exemplo de código-fonte para a implementação de um módulo.

- A classe `Theme`, também ela um módulo do tema e que representa o micro-núcleo propriamente dito. Esta classe é responsável por registar a pasta `thememodules` como uma pasta de biblioteca de funções, carregar todos os ficheiros com um nome que termine com o sufixo `.load.php` dos sub-diretórios da pasta `thememodules` (carregando os ficheiros de inicialização e de configuração dos módulos), configurar o suporte a múltiplos idiomas e carregar os ficheiros de idioma a partir da pasta `langs` do tema base e dos temas específicos (se a execução estiver a ocorrer a partir deste contexto), indicar ao WordPress que o tema suporta funcionalidades HTML5 nos formulários de pesquisa, nas galerias e nas legendas, e fornecer indicações ao WordPress para carregar o ficheiro de estilos do tema. Esta classe regista ainda um

método próprio com o sistema de carregamento automático de ficheiros do PHP baseado no nome de classes desconhecidas, permitindo o carregamento de módulos com base no nome das classes que os definem.

- Invocação do método de inicialização da classe `Theme` e que irá inicializar o micro-núcleo.

Para que o núcleo seja inicializado o ficheiro `functions.php` contém como única instrução a indicação para carregamento do ficheiro `thememodules/ThemeModules.php` que irá autonomamente iniciar o micro-núcleo e o carregamento de todos os módulos que compõem o tema (Anexo XXXII). O ficheiro `functions.php` não carece de qualquer instrução adicional pois toda a funcionalidade está encapsulada nos módulos carregados pelo núcleo e que passam a assumir o controlo pleno do funcionamento da solução.

Os restantes ficheiros disponíveis na raiz do tema, controlam aspetos gráficos e funcionais específicos:

- `404.php` – contém o modelo a apresentar para a página de recurso não encontrado.
- `category.php` – contém o modelo para as páginas de categorias, como por exemplo, as categorias de notícias.
- `comments.php` – modelo da página de comentários. No sentido de reduzir o esforço associado à moderação de comentários na solução, esta funcionalidade foi desativada neste ficheiro.
- `header-base.php` – modelo específico deste tema para o componente de cabeçalho apenas com a estrutura essencial de uma página HTML e sem qualquer conteúdo visual.

- `footer-base.php` – modelo específico deste tema para o componente de rodapé sem elementos visuais. Basicamente este componente é utilizado apenas para terminar as *tags* que foram iniciadas no componente `header-base.php`.

- `header.php` – modelo para o componente com o cabeçalho principal do tema, com a identidade gráfica, barra de topo, menu, etc.

- `footer.php` – modelo para o componente com o rodapé principal do tema que, para além de terminar as *tags* que foram iniciadas no componente `header.php`, pode inserir os conteúdos associados ao rodapé do tema (se estes estiverem definidos).

- `index.php` – ficheiro com o modelo por omissão para conteúdos aos quais não foi possível associar um outro modelo mais específico (Anexo XXXIII). Com recurso à API do WordPress faz uso dos componentes principais de cabeçalho e rodapé descritos anteriormente, bem como do modelo descrito no ficheiro `loop.php`.

- `loop.php` – ficheiro com o modelo para o ciclo principal associado ao pedido efetuado pelo cliente (Anexo XXXIV). Sempre que é efetuado um pedido a um site WordPress, essa solicitação é transformada numa pesquisa à base de dados que é depois traduzida num ciclo. Através deste ciclo, um tema pode utilizar expressões condicionais e de repetição para verificar se o pedido devolveu algum recurso e, se sim, e enquanto existirem recursos associados ao pedido, o sistema deve carregar um recurso em cada iteração e apresentar o modelo descrito neste modelo para esse bloco de conteúdo.

- `navigation.php` – modelo para o componente de paginação, caso o pedido tenha devolvido vários recursos para os quais seja necessário apresentar uma ferramenta visual de paginação para facilitar a navegação.

- `screenshot.jpg` – captura de ecrã com a representação visual do aspeto do tema. Esta imagem é utilizada na área administrativa para facilitar a identificação visual de um

tema no seletor de temas.

- `search.php` – modelo utilizado para a apresentação de resultados de pesquisa.

- `searchform.php` – modelo do componente de pesquisa do tema.

- `sidebar.php` – modelo para a barra lateral do tema. Neste projeto este componente foi desativado, dado que a identidade visual não pressupõe a existência deste tipo de zonas, tal como estão previstas pela API do WordPress.

- `site-info.php` – modelo específico deste tema, utilizado para apresentar informação adicional (a partir de um dos módulos desenvolvidos no âmbito deste projeto) e que pretende apresentar instruções de apoio à navegação sobre o *site* para visitantes que utilizem tecnologias de apoio, tais como leitores de ecrã.

- `site-map.php` – modelo específico deste tema, utilizado para a apresentação do mapa do *site*.

- `site-technical-sheet.php` – modelo específico deste tema, utilizado para a apresentação da ficha técnica do *site*.

4.7.2. Estrutura dos temas específicos

Os temas específicos permitem isolar funcionalidades próprias associadas aos requisitos de um determinado projeto, encapsulando as mesmas num tema exclusivo desse projeto e que estende as funcionalidades herdadas a partir do tema base. Dado que a maioria das funcionalidades são obtidas a partir do tema base, estes temas específicos acabam por ser bastante minimalistas. As principais diferenças destes temas centram-se nos ficheiros de meta-dados (com a informação específica de cada um dos temas), nos módulos específicos em cada um dos temas e nos ficheiros com os modelos específicos que o programador deseje

reimplementar.

- Pasta `css` – os ficheiros desta pasta importam algumas definições do tema base (tais como valores de variáveis, ou instruções CSS específicas), podendo as mesmas ser sobrepostas nos temas específicos consoante as necessidades.

- Pasta `langs` – pacote de idioma específico do tema filho e que é unido ao pacote de idioma do tema base.

- Pasta `thememodules` – contém instruções de inicialização e carregamento do micro-núcleo no ficheiro `ThemeModules.php` (Anexo XXXV) e respetivos módulos específicos do tema.

- Os ficheiros `.gitattributes`, `.gitignore`, `.Gruntfile.js` e `functions.php` apresentam o mesmo conteúdo e desempenham as mesmas funções que as descritas na secção 4.7.1 relativamente ao tema base.

- Os ficheiros `README.md`, `package.json`, `screenshot.jpg` e `style.css` apresentam as versões atualizadas dos meta-dados relativos ao tema específico, desempenhando funções semelhantes aos seus congéneres do tema base.

- Todos os restantes ficheiros dizem respeito a reimplementações de modelos existentes no tema base ou novos modelos relativos a funcionalidades específicas do tema. Por exemplo, é possível introduzir um cabeçalho completamente novo num tema específico sobrepondo o ficheiro `header.php` com as alterações pretendidas. Neste cenário o WordPress irá utilizar a versão específica deste ficheiro ignorando a modelo de cabeçalho existente no tema base.

4.8. Módulos globais desenvolvidos

No âmbito deste projeto foram desenvolvidos cerca de 40 módulos que implementam funcionalidades comuns a todos os *sites* desenvolvidos, através dos temas específicos que estendam a funcionalidade a partir do tema base. Estes módulos tratam de questões como suporte a múltiplos idiomas, criação de mapas do *site*, questões relacionadas com SEO, gestão de menus, entre muitas outras funcionalidades que serão descritas individualmente nesta secção. Estes módulos comuns são disponibilizados na pasta `thememodules` do tema base, devidamente encapsulados em sub-diretórios, que são depois disponibilizados de forma transparente aos temas especializados a partir do tema base.

4.8.1. TinyMCE

O TinyMCE[23] é um editor de texto rico, baseado em JavaScript e HTML, que permite converter alguns elementos de páginas HTML (tais como áreas de texto, ou elementos `div`) em editores visuais nos quais é possível criar e gerir código HTML de um modo gráfico e onde se espera que o resultado apresentado no editor seja o resultado final que será apresentado ao utilizador (WYSIWYG) após a publicação do conteúdo. É o editor por omissão para áreas de texto com conteúdos HTML do WordPress, mas utilizado em muitas outras plataformas tais como o Drupal, Joomla!, Django, etc. É uma ferramenta bastante completa com uma API própria e exposta através da própria API do WordPress.

Dada a importância deste componente na edição de conteúdos no sistema, foi desenvolvido um módulo próprio que permite adaptar a ferramenta às necessidades específicas do projeto, através do desenvolvimento de uma API intermédia que os restantes módulos do tema podem utilizar para adicionar funcionalidades ao editor. Assim, este módulo é responsável por:

23 https://github.com/tinymce/tinymce

- Remover a inclusão automática de parágrafos dos campos de conteúdo HTML e excerto de conteúdo, funcionalidade ativada por omissão pelo WordPress mas que em alguns contextos pode gerar código HTML inválido.

- Adicionar os estilos de conteúdos específicos do tema (ficheiro `css/editor-style.css`) à área de edição de conteúdo.

- Adicionar um botão ao editor no qual outros módulos possam adicionar funcionalidades específicas.

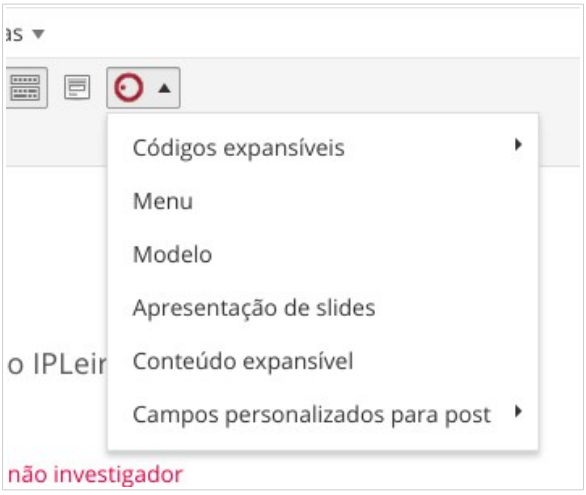


Figura 4.7: Botão e menu com funções específicas dos módulos

- Permitir que outros módulos possam adicionar estilos adicionais o editor.

- Adicionar suporte ao HTML5, bem como a redefinição de atributos adicionais tais como `role`, `aria-controls`, `aria-expanded`, etc., importantes para questões relacionadas com a acessibilidade (previstos na norma WAI-ARIA, mas que não eram nativamente suportados pelo TinyMCE) e forçar a formatação de código HTML nesta vista de edição.

- Fornecer uma API baseada em filtros através da qual outros módulos possam definir elementos HTML personalizados para implementação de funcionalidades de edição avançadas.

- Disponibilização de uma API para registo de novas extensões para o TinyMCE a partir de outros módulos.

4.8.2. ActionExecutionContentFilter

O WordPress disponibiliza uma API de códigos expansíveis/curtos (*shortcodes*[24]) que permite transformar códigos como `[hello subject="World"]` em `Hello World` (apenas um exemplo). Com recurso a esta funcionalidade extensões ou temas podem registar um código curto, indicar quais são os atributos suportados e qual a função que deve ser executada para processar o código registado. Sempre que o WordPress encontrar este código numa área de conteúdo, processa o mesmo e envia os valores dos parâmetros fornecidos à função registada, permitindo que a função gere um conteúdo específico que será substituído pelo código do *shortcode* originou a invocação.

Este módulo recorre a esta API para registar um código curto com o identificador `ActionExecute` com suporte aos parâmetros `action` e `args`, como no Exemplo 1.

```
[ActionExecute action="courses-taxonomy-title" args="title=Cursos"]
```

Exemplo 1: Exemplo de um *shortcode* gerido pelo módulo ActionExecutionContentFilter

No exemplo anterior, quando o WordPress se depara com um código deste tipo (por exemplo, ao processar o conteúdo de uma página a devolver a um visitante ao nível do *front-end*), irá invocar o método registado por este módulo para este *shortcode* que analisará o nome a ação solicitada e executará as funções associadas à mesma enviando os parâmetros recebidos

24 https://codex.wordpress.org/Shortcode_API

através do parâmetro `args`. Isto permite que um qualquer módulo associe um método próprio a uma qualquer ação (como a ação `courses-taxonomy-title` do exemplo), que será executada sempre que o WordPress encontrar uma referência a esta ação, permitindo qualquer módulo tirar partido da programação orientada a eventos para introduzir código específico em momentos muito concretos do fluxo de execução, sem necessidade de definir os seus próprios códigos curtos adicionais.

Este módulo faz ainda uso da API do módulo TinyMCE para adicionar uma opção ao menu do editor que insere um código exemplo com o formato esperado para o *shortcode*.

4.8.3. Bar

O principal objetivo deste módulo é o da personalização da barra de edição fornecida pelo WordPress (que é apresentada no topo dos *sites*), ocultando a mesma na área pública e adicionando a imagem institucional do projeto a este componente na área administrativa.

4.8.4. Branding

O módulo de *Branding* é responsável pela definição da marca/imagem institucional dos temas dos quais faz parte. É responsável por personalizar a página de autenticação, alterando o logótipo para a imagem institucional do projeto, bem como o respetivo texto e endereço. É também este módulo que adiciona no cabeçalho de todas as páginas geradas as várias versões dos ícones apresentados nos navegadores Web dos clientes (*favicon*).

4.8.5. Colorbox

Colorbox é uma extensão jQuery que permite a criação de janelas modais para apresentação de informação num formato destacado. É utilizada para, por exemplo, apresentar imagens na

Web no tamanho máximo disponível na área de visualização da mesma a partir de miniaturas numa página. Este módulo adiciona suporte a esta biblioteca no sistema permitindo que, qualquer hiperligação com um endereço de uma imagem associada a uma imagem (por exemplo uma miniatura da imagem), possa destacar a imagem referenciada na hiperligação, apresentando a mesma numa janela modal com um tamanho adequado que permita ver a mesma com mais pormenor.

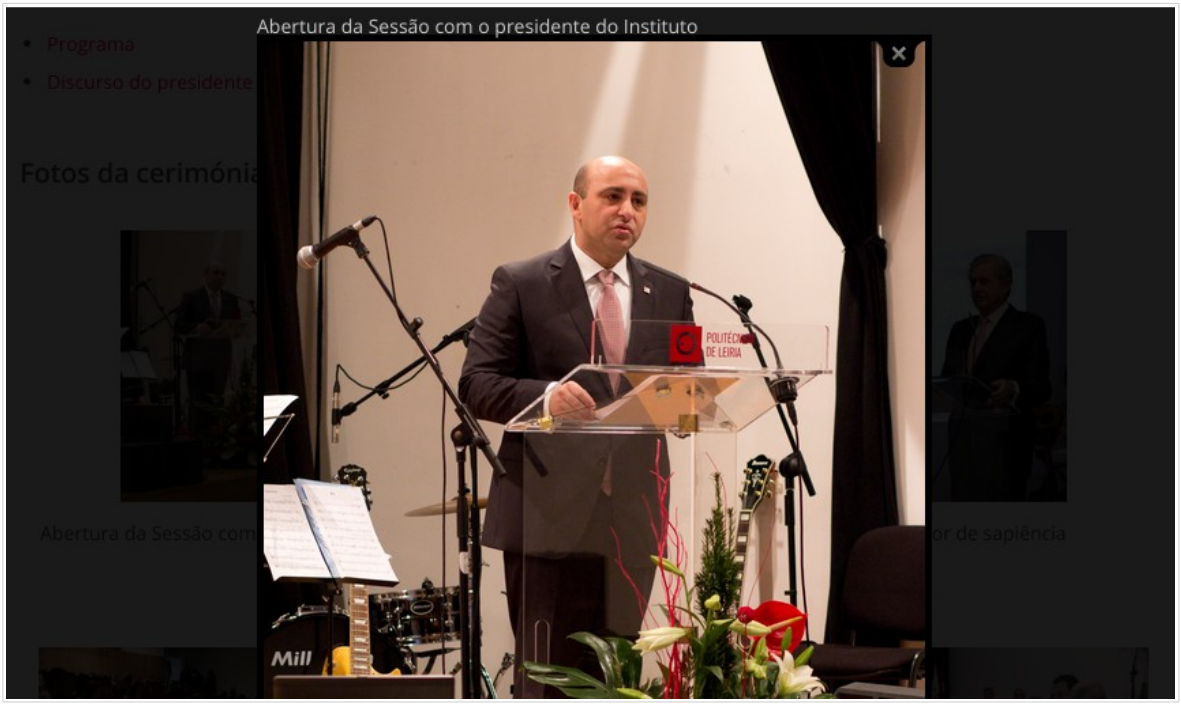


Figura 4.8: Exemplo de uma janela modal criada pelo Colorbox no sistema

No exemplo da Figura 4.8, é apresentada uma versão destacada de uma imagem com um tamanho razoável a partir da interação com uma versão mais reduzida de uma outra imagem disponibilizada na área de conteúdo. Ao clicar na versão reduzida da imagem o visitante teve acesso a uma imagem com maior definição através de uma janela modal criada pela biblioteca Colorbox integrada no sistema.

4.8.6. ContentMenu

Os menus são uma parte fundamental de qualquer *site* Web. O WordPress introduziu a partir da versão 3.0 um sistema de gestão de menus independente da organização interna dos conteúdos. Com recurso à API disponibilizada, os temas devem informar se suportam este novo sistema e quais as zonas na estrutura do tema onde podem ser adicionados menus. Depois o administrador do *site* poderá gerir os menus adicionando, organizando e removendo elementos aos menus que são depois associados por este às respetivas áreas definidas no tema.

Nos temas desenvolvidos no âmbito deste projeto apenas foi definida uma área específica no cabeçalho dos temas onde pode ser inserido o menu principal de cada um dos *sites*. No entanto a proposta gráfica pressupunha que pudessem existir vários locais distintos no conteúdo das páginas onde pudessem ser inseridos menus específicos. Como não existe uma estrutura rígida comum a todas as páginas, com áreas bem definidas para a colocação de menus que pudessem ser indicadas ao WordPress, optou-se por desenvolver este módulo cuja principal função é, com recurso a um *shortcode* específico permitir a inserção de um qualquer menu disponível na área administrativa numa zona de conteúdo de uma determinada página, artigo ou qualquer outro tipo de conteúdo mais específico de um *site*. Isto permite dar ao gestor de conteúdos um controlo absoluto sobre a apresentação dos menus, podendo definir quais os menus que pretende ver apresentados, em que recursos e em que zona no conteúdo.

```
[content_menu menu="submenu-pt" container_class="nav noicon centered-menu vertical nocollapse"]
```

Exemplo 2: Exemplo de um *shortcode* para a inserção de um menu na área de conteúdo

Tecnicamente este módulo utiliza o *shortcode* com o nome `content_menu` para gerar o menu identificado com o nome, ID ou *slug* através do atributo `menu`. Suporta praticamente todos os atributos que são utilizados como parâmetros da função `wp_nav_menu`[25] da API de menus do WordPress permitindo assim definir contentores HTML para o menu, classes CSS, formatos

25 https://codex.wordpress.org/Function_Reference/wp_nav_menu

internos dos elementos do menu, entre outros parâmetros que definem o aspeto e comportamento dos menus inseridos.

Este módulo faz uso pleno do módulo TinyMCE para adicionar uma ferramenta visual para a inserção ou edição dos menus com recurso a esta ferramenta.

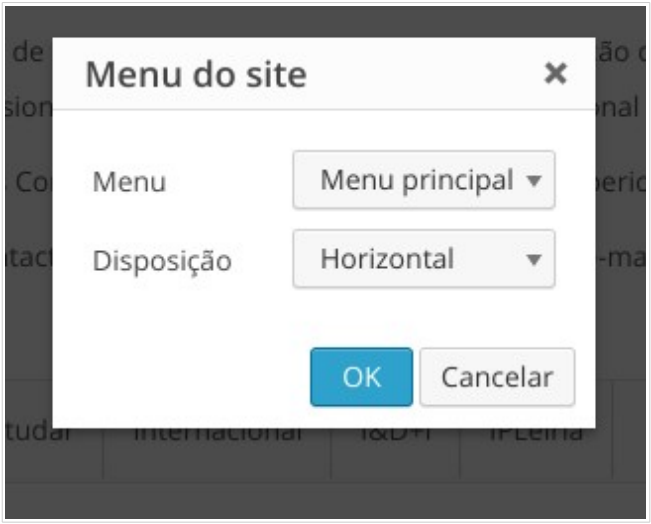


Figura 4.9: Assistente de inserção de menus

Este assistente permite selecionar o menu, dos menus previamente criados na área administrativa, que o utilizador pretende inserir na área de conteúdo, no local onde o cursor foi colocado. Para além da seleção do menu, o utilizador pode ainda especificar qual a disposição que o mesmo deve assumir, com base nas duas propostas de funcionamento de menus previstas pelo *designer*.

Para que o utilizador tenha uma noção mais real do aspeto final da página com o menu selecionado, o sistema gera a pré-visualização do menu após a inserção do mesmo na área de conteúdo, como se pode ver no exemplo da Figura 4.10.

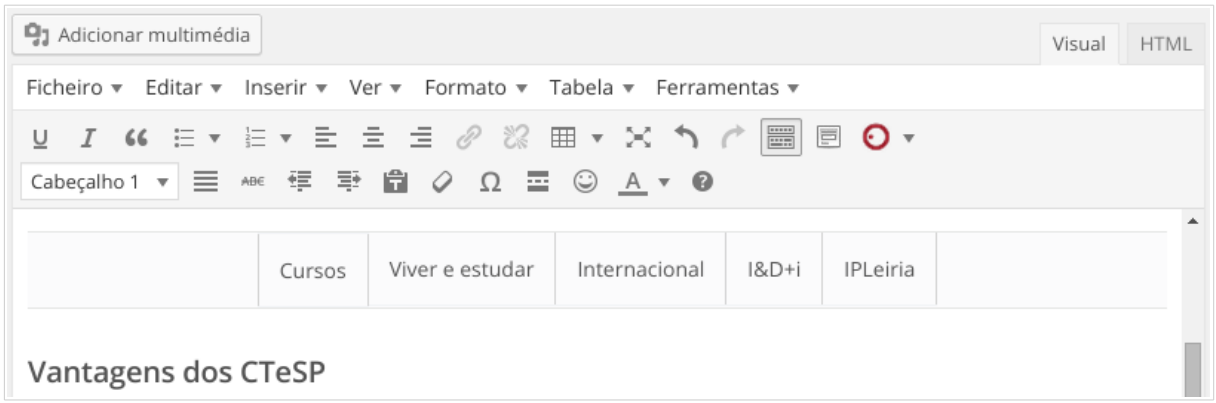


Figura 4.10: Pré-visualização do menu na área de edição do conteúdo

No entanto, apesar de visualmente estar a ser gerado o código HTML para a correta representação do menu, internamente o módulo continua garantir que a representação no formato *shortcode* é mantida, assegurando assim o correto funcionamento do sistema e garantindo que se o menu for atualizado, estas alterações são refletidas de forma transparente para os utilizadores do sistema.

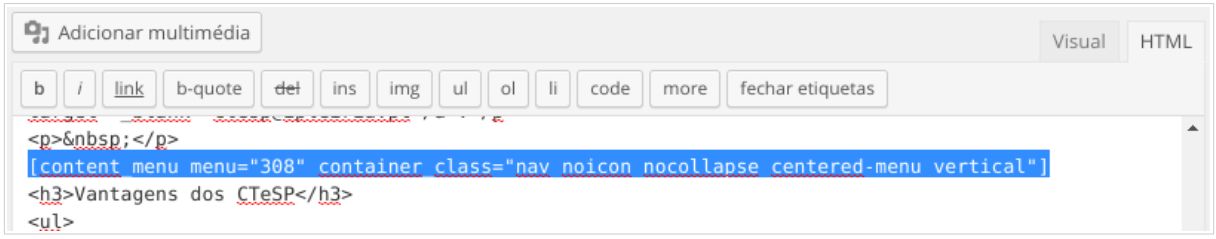


Figura 4.11: Código que representa o menu na vista de código do editor WYSIWYG

4.8.7. ContentTemplates

A flexibilidade tem, também ela, o seu custo. O facto de não existir uma estrutura rígida que mantenha o aspeto gráfico entre todas as páginas do sistema pode implicar um esforço significativo de gestão para as páginas que apresentam uma estrutura comum com outras

páginas do sistema. Este módulo pretende fornecer uma ferramenta para que permita facilitar esta gestão através da introdução de uma funcionalidade denominada por "modelos de conteúdo". Estes modelos de conteúdo permitem definir um conjunto de modelos base na área de administrativa que podem ser utilizados para replicar estruturas previamente criadas no sistema. Assim, um administrador ou gestor de conteúdos pode criar uma estrutura de página através de um destes modelos e reutilizar este modelo sempre que dele necessitar para recriar páginas com estruturas idênticas.

Este módulo apresenta dois modos de funcionamento distintos. No primeiro, o utilizador cria os modelos com a estrutura pretendida e, sempre que cria uma nova página onde pretenda replicar essa estrutura, utiliza a opção `Modelo` do menu específico do projeto no editor WYSIWYG para selecionar o modelo cujo conteúdo pretende inserir na área de conteúdo.

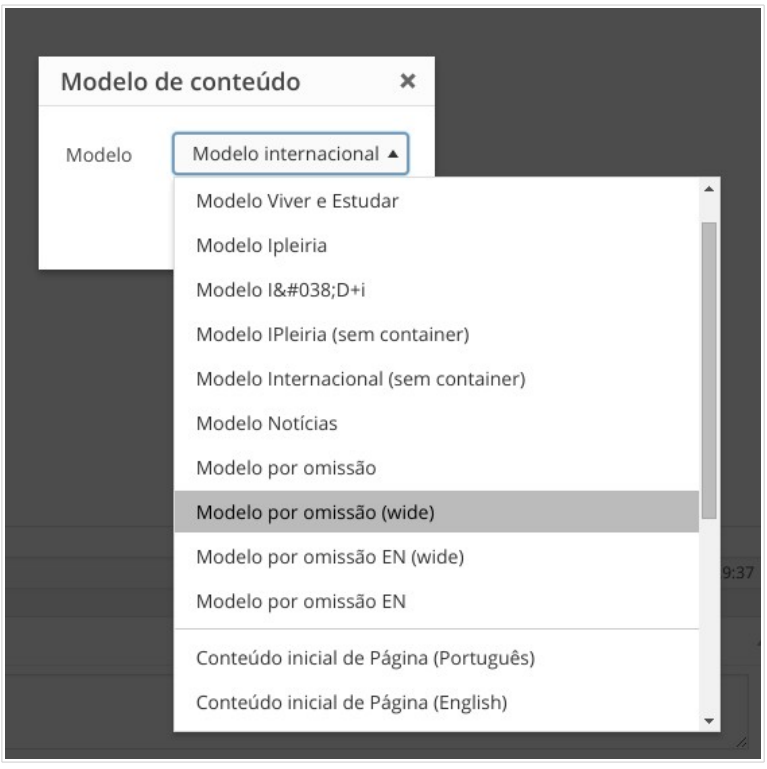


Figura 4.12: Interface de seleção de modelos de conteúdo

Nesta abordagem o modelo selecionado é inserido na área de conteúdo, funcionando como o conteúdo inicial do documento cujo gestor de conteúdos poderá personalizar sem restrições, permitindo que a estrutura seja completamente alterada em função das necessidades.

No entanto, neste primeiro cenário, se o modelo for alterado, estas alterações não são refletidas nas páginas que adotaram a estrutura definidas antes da aplicação das alterações. Neste contexto, este módulo introduz um segundo modo de funcionamento onde o editor pode selecionar qual o modelo base de uma determinada página ou recurso através da caixa de opções criada pelo módulo no interface de edição de conteúdos.

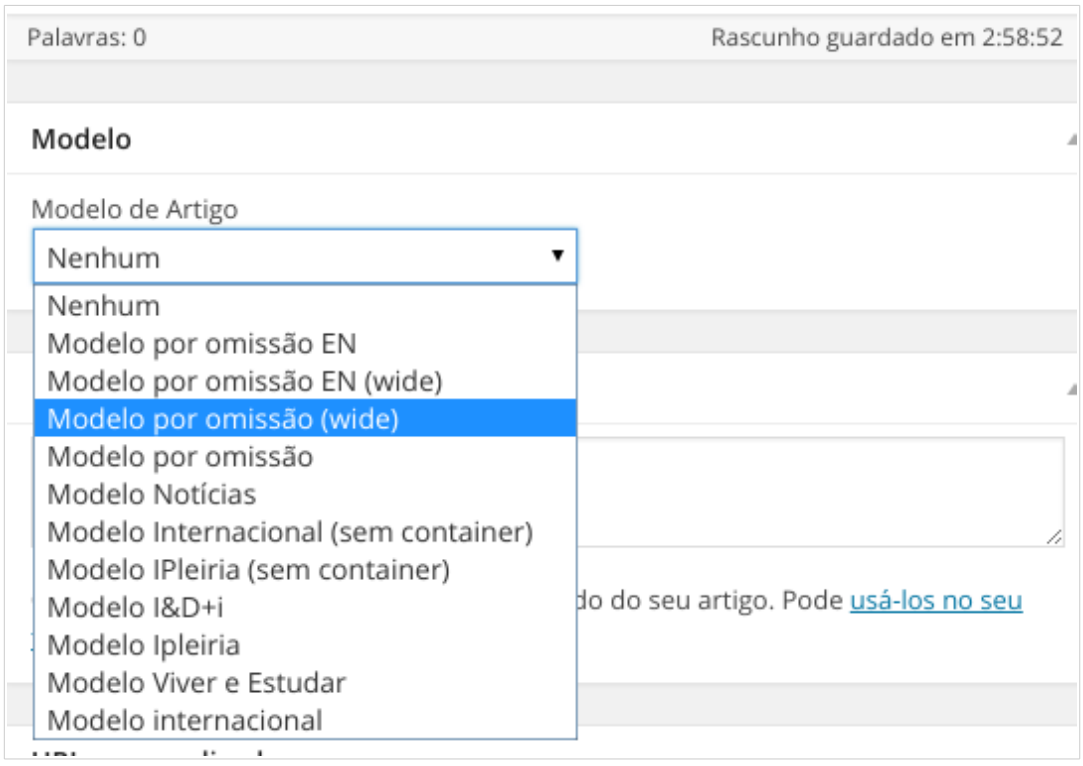


Figura 4.13: Interface de seleção de modelos de conteúdo base de um recurso

Neste cenário o modelo não é inserido diretamente na área de conteúdo mas é o conteúdo do recurso que é inserido numa zona definida no modelo no momento em que a página está a ser gerada. Para o efeito, deve ser utilizado um código especial (`[_ued-main-content]`) no

conteúdo do modelo que indica o ponto exato onde o conteúdo deverá ser inserido no modelo.

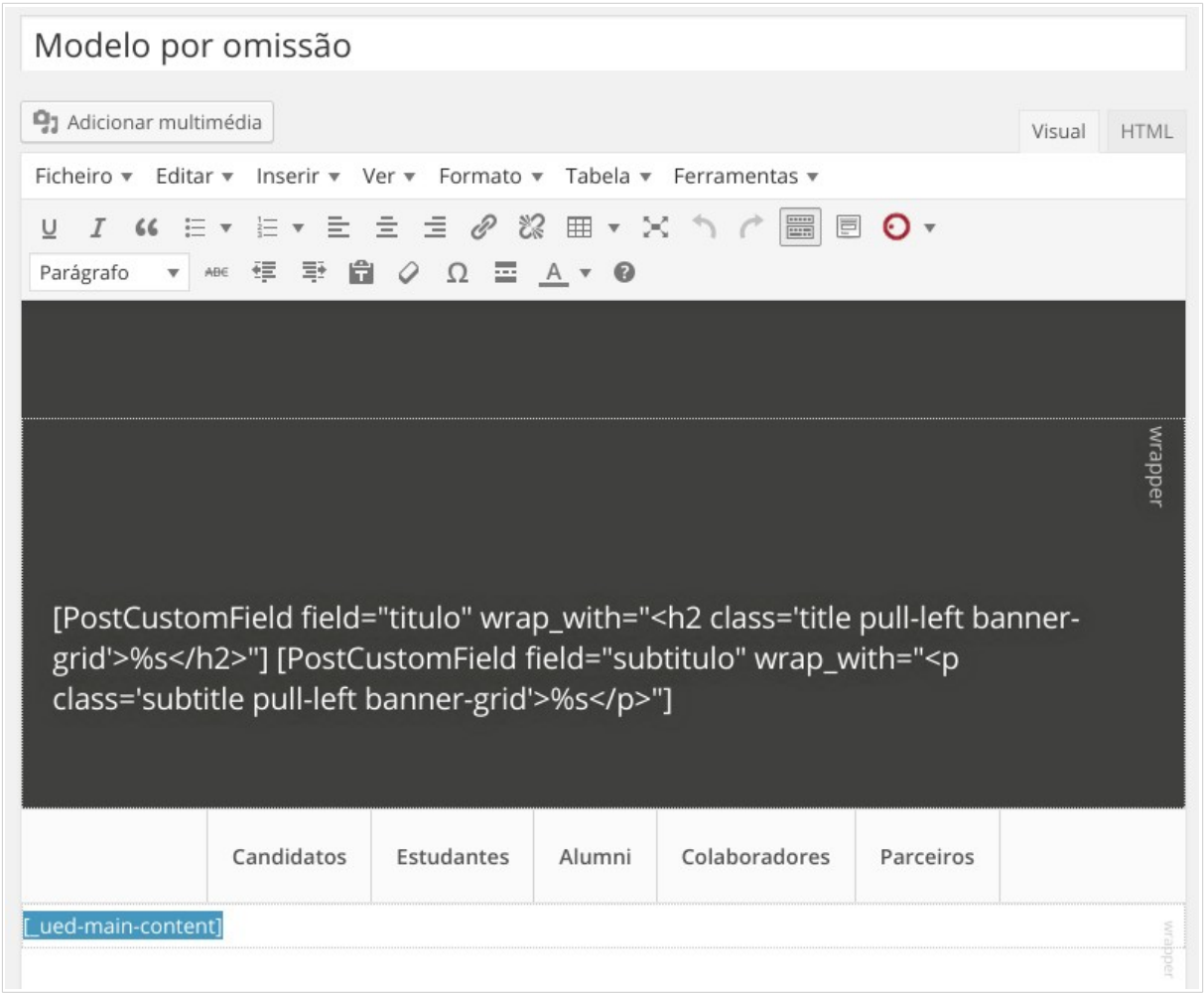


Figura 4.14: Exemplo de um modelo base

Para além da criação de uma secção para gestão destes modelos, este módulo disponibiliza ainda uma área para a definição de modelos iniciais por omissão para os tipos de conteúdo existentes no sistema. Isto permite que um administrador possa definir que todas as páginas que vão ser criadas no sistema devem ter uma determinada estrutura inicial que depois poderá ser editada pelos gestores de conteúdo. Assim, em vez de uma página ou um artigo ter a área de conteúdo em branco, poderá ter um conteúdo inicial com a estrutura pré-definida para esse

tipo de conteúdo. Para além desta funcionalidade, o módulo disponibiliza ainda uma API que pode ser utilizada para outros módulos para criar secções adicionais com outros modelos específicos, para além dos fornecidos por este módulo.

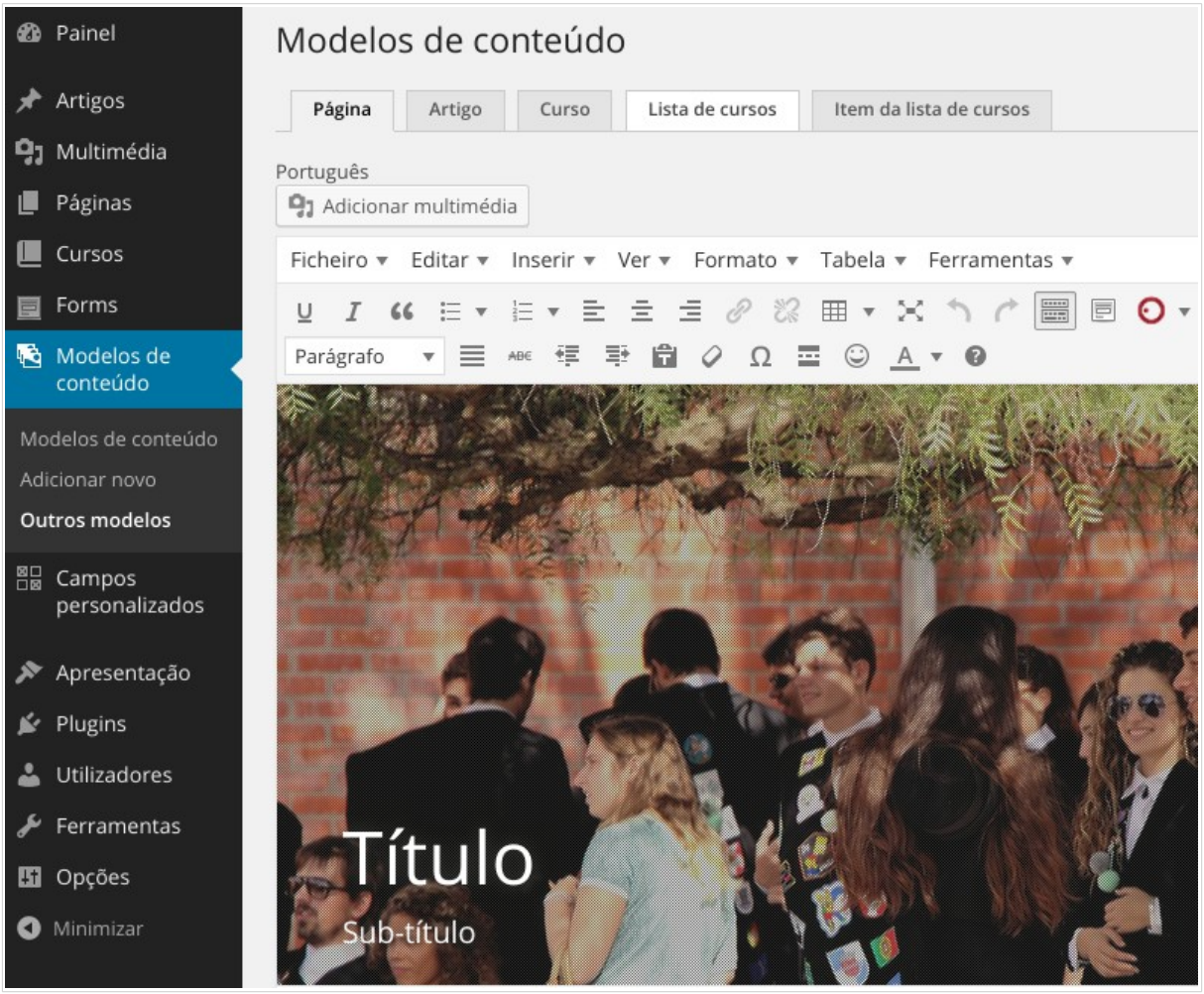


Figura 4.15: Interface com os modelos de sistema adicionais

Por exemplo, na Figura 4.15, para além dos modelos `Página`, `Artigo` e `Curso` foram também adicionados os separadores que permitem definir os modelos para a `Lista de cursos` e o `Item da lista de cursos`, utilizados em contextos muito específicos pelo sistema de acordo com as definições do módulo que os criou.

SGCPI

Tecnicamente é um módulo com alguma complexidade, mas que elimina algumas das necessidades de implementação de código específico nos próprios temas para a criação de modelos específicos, virtualizando para a área administrativa estas definições e flexibilizando considerável a criação deste tipo de modelos.

4.8.8. Cycle

Este módulo adiciona as funcionalidades da extensão jQuery Cycle2[26] ao sistema, permitindo criar transições animadas (*slideshows*) entre elementos gráficos, tais como imagens, vídeos ou outros elementos multimédia.

```
[cycle_slideshow
data_cycle_fx="scrollHorz" data_cycle_pause_on_hover="true"
swipe="true" timeout="6000"]
      <img src="http://malsup.github.io/images/p1.jpg" alt="" />
      <img src="http://malsup.github.io/images/p2.jpg" alt="" />
      <img src="http://malsup.github.io/images/p3.jpg" alt="" />
      <img src="http://malsup.github.io/images/p4.jpg" alt="" />
[/cycle_slideshow]
```

Exemplo 3: Exemplo de um *shortcode* para a inserção de um *slideshow*

Quando inserido no conteúdo de uma página, o *shortcode* do Exemplo 3, permite criar uma transição entre as quatro imagens definidas no conteúdo do elemento, como uma transição horizontal, onde a animação é suspendida quando o cursor é colocado sobre o elemento, que permite a transição entre imagens com o gesto de arrastamento e onde a transição entre duas imagens é efetuada em intervalos de 6 segundos. O *shortcode* suporta as várias opções previstas na API do Cycle2[27] sob a forma de atributos adicionais (onde o caráter `-` deve ser substituído pelo caráter `_` no sentido de garantir a compatibilidade com a API de *shortcodes* do WordPress).

26 http://jquery.malsup.com/cycle2/

27 http://jquery.malsup.com/cycle2/api/#options

Este módulo faz um uso intensivo da API do módulo TinyMCE para criar um interface para a auxiliar a inserção deste tipo de elementos. Através da opção `Apresentação de slides` no menu do sistema no editor WYSIWYG, é apresentado um assistente que permite selecionar opções como tipo de transição, intervalo de transição, conteúdo inicial, entre outras opções.

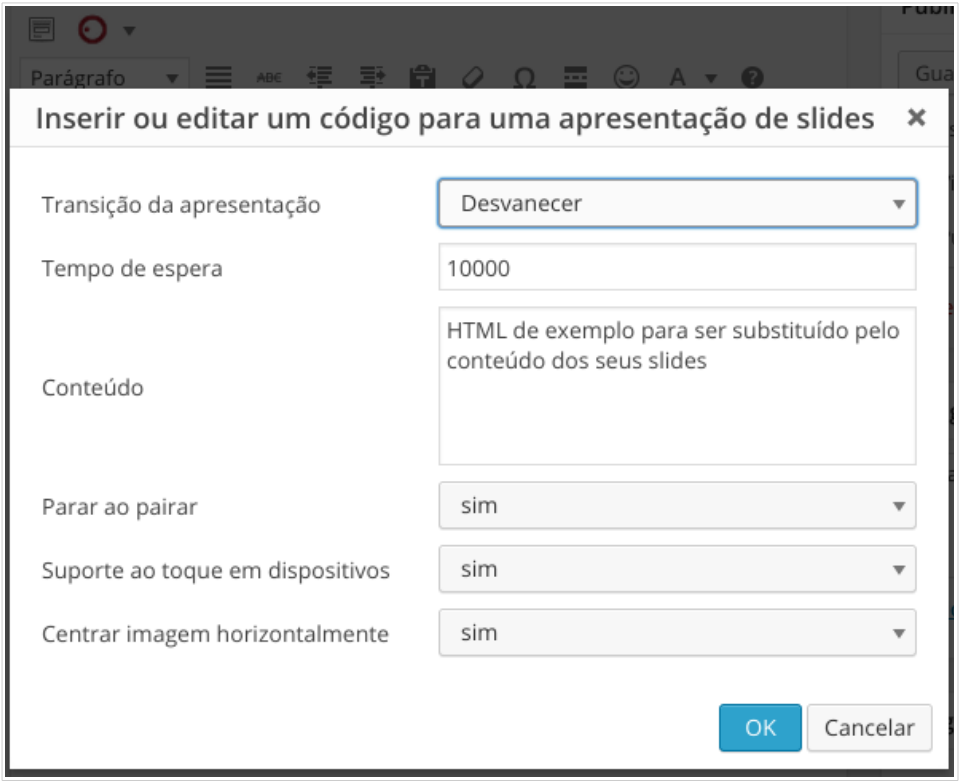


Figura 4.16: Assistente de inserção de *slideshows*

Após ser inserido, é possível editar visualmente o elemento, deslocando o cursor sob o mesmo e clicando no botão de edição fornecido pelo sistema.

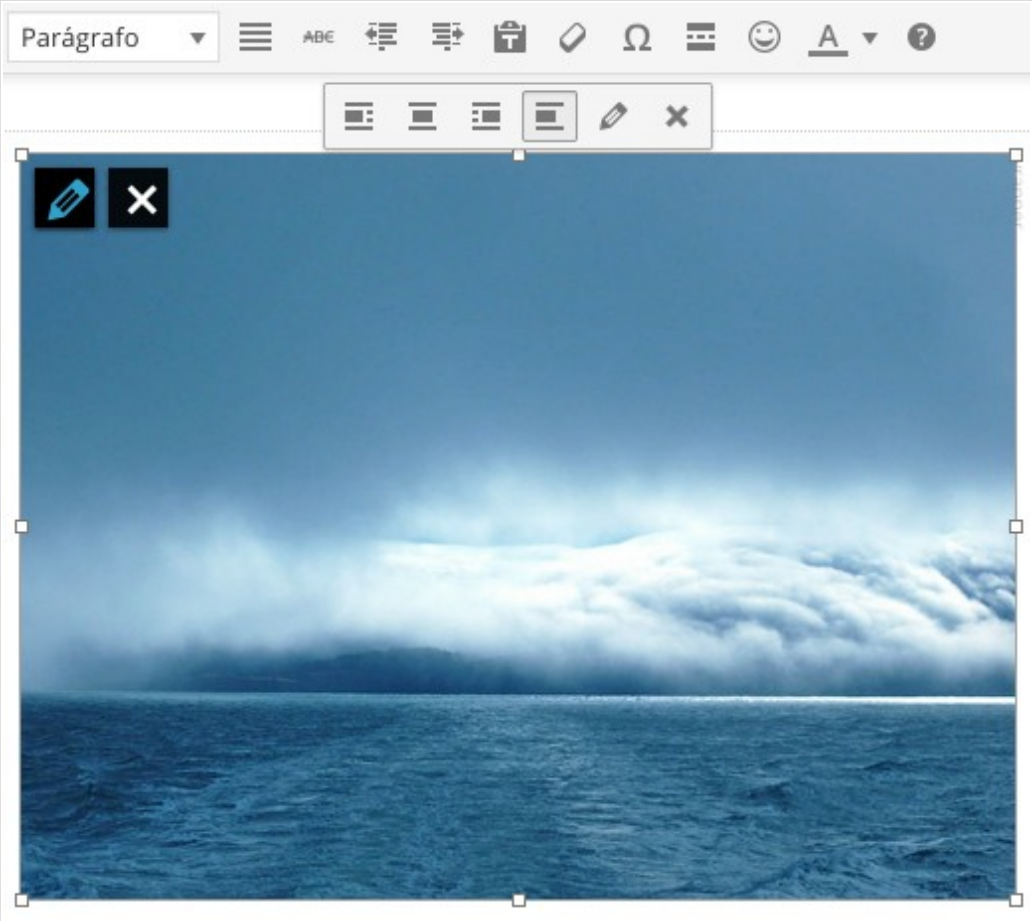


Figura 4.17: Opção de edição de *slideshows*

Ao ativar o modo de edição para o elemento, este é expandido sendo apresentados todos os conteúdos que sejam parte integrante do mesmo. O editor poderá depois remover ou adicionar elementos clicando no local ou no elemento que pretende editar, utilizando as ferramentas disponibilizadas pelo sistema. Para que as alterações sejam aplicadas, basta posicionar o cursor numa zona fora do elemento, para que seja novamente ativado o modo e pré-visualização por parte do módulo.

Nos navegadores suportados, o sistema disponibiliza uma pré-visualização ativa no modo de edição. Isto significa que é apresentada uma pré-visualização da transição na vista visual do

editor com o aspeto que terá na área pública. À semelhança do que acontece com outros módulos, este sistema também faz a tradução de forma transparente entre o código HTML utilizado na vista visual e o *shortcode* que será armazenado em conjunto com o conteúdo na base de dados, garantindo o correto funcionamento do sistema.

4.8.9. ExpandableContent

As propostas gráficas deste projeto seguem uma linha gráfica fortemente inspirada em *single page design*, onde se minimiza o número de páginas em função do aumento da sua extensão. Isto acarreta alguns riscos, nomeadamente do ponto de vista da usabilidade, onde o excesso de informação pode alienar o utilizador. Uma das técnicas que se utilizou para minimizar os efeitos deste problema foi através da criação de zonas de conteúdo que podem ser expandidas ou colapsadas, permitindo que o utilizador possa aceder ou ocultar informação de acordo com as necessidades.

Mais uma vez, com recurso à API de *shortcodes* do WordPress e a código JavaScript desenvolvido, foi possível implementar um elemento que permite encapsular conteúdo em *shortcodes* identificados pela *tag* `ExpandableContent` e que permite criar elementos no *front-end* onde esse conteúdo pode ser ocultado ou apresentado, interagindo com o título do elemento.

Tecnicamente este módulo suporta três atributos ao nível do *shortcode*:

- `title` – Parâmetro obrigatório que define o título a apresentar e com o qual o utilizador poderá interagir para expandir ou colapsar o elemento.
- `expanded` – Parâmetro opcional que indica o estado inicial do elemento, onde `false` indica que o mesmo estará inicialmente colapsado e `true` para que esteja expandido. O valor por omissão é `false`.
- `id` – Parâmetro opcional que define um identificador específico para o elemento. O valor por omissão é um identificador único gerado aleatoriamente.

SGCPI

Na implementação deste elemento houve preocupações claras com as questões de acessibilidade, garantindo que o estado do elemento é convenientemente atualizado através dos atributos previstos no WAI-ARIA 1.0, fornecendo informações adequadas a ferramentas como leitores de ecrã.

Mais uma vez foi também utilizado o módulo TinyMCE para inserção de uma funcionalidade que permite a inserção de um código exemplo para este elemento, através da opção `Conteúdo expansível` no menu específico disponibilizado no editor.

4.8.10. Favicon

Com a introdução de funcionalidades como o suporte a separadores nos navegadores Web, o ícone de favoritos ganhou ainda mais relevância do ponto de vista da usabilidade, permitindo que o utilizador consiga identificar facilmente qual o *site* que está a ser apresentado num determinado separador, através deste pequeno elemento gráfico. Em alguns dispositivos móveis este ícone é utilizado para decorar os atalhos que tenham sido criados para *sites*, pelo que é conveniente garantir que o mesmo tem um aspeto gráfico apelativo, que facilite a identificação visual do *site* que representam.

Este módulo é responsável por fornecer um acesso direto, a partir do endereço base do *site*, a várias versões deste recurso apontando para as respetivas versões disponíveis no tema. Assim, assumindo um *site* com o endereço base http://www.ipleiria.pt, se um cliente solicitar o recurso http://www.ipleiria.pt/favicon.ico este módulo irá capturar esse pedido e devolver a imagem disponível na pasta do tema no caminho `[pasta do tema]/img/favicon/favicon.ico`. Situação semelhante acontece para os ficheiros `favicon_32x32.png`, `favicon_57x57.png`, `favicon_72x72.png`, `favicon_114x114.png`, `favicon_144x144.png`.

Para além de enviar o conteúdo dos ficheiros propriamente ditos, o módulo faz ainda a

identificação do tipo, cálculo do tamanho e cálculo da *hash* do ficheiro, fornecendo estas informações sob a forma de cabeçalhos HTTP, permitindo que o conteúdo possa ser armazenado em *cache*.

4.8.11. FormManagerExtraFields

Os formulários são uma peça fundamental em projetos Web que promovam uma interação com os seus visitantes. Este projeto previa a criação de uma série de formulários em várias áreas do *site*, desde formulários de contacto, a formulários de requisições, esclarecimento de dúvidas, entre outros. Neste sentido é importante garantir que esta funcionalidade é fornecida por uma ferramenta sólida, simples de utilizar e capaz de fornecer uma resposta eficaz à criação de formulários.

Verificou-se que existem várias soluções no universo de extensões do WordPress que permitem criar este tipo de recursos pelo que, após uma análise cuidada, selecionou-se a extensão Form Manager[28] como solução tecnológica para criação de formulários no sistema. Esta extensão tem mais de 240 mil instalações e está muito bem classificada no grupo de extensões que fornecem este tipo de funcionalidades. Suporta vários tipos de campos de dados, desde campos de texto, áreas de texto, listas pendentes, botões rádio, caixas de verificação, envio de ficheiros, reCAPTCHA, entre outros. Como funcionalidades disponibiliza mecanismos de validação de dados, definição de campos obrigatórios, mensagens de correio eletrónico personalizadas, modelos para formulários, exportação de dados no formato CSV e uma API que permite algum grau de personalização. A construção de formulários é feita com recurso a ferramentas visuais bem integradas no interface da plataforma, cujos testes com utilizadores mostraram ser simples e intuitivas. É uma extensão gratuita sem serviços *premium* associados.

Apesar da ferramenta fornecer um serviço bastante competente na gestão e apresentação de

28 https://wordpress.org/plugins/wordpress-form-manager/

formulários, as ferramentas automáticas de validação de acessibilidade reportaram uma série de não conformidades relativamente às recomendações WCAG 2.0. Este módulo introduz três sub-módulos que recorrem à API da extensão para disponibilizar novos tipos de campos para substituir os tipos de campos disponibilizados pela extensão, onde se tentou resolver os problemas de acessibilidade identificados:

- Campo de texto melhorado (`BetterTextField`);
- Área de texto melhorada (`BetterTextArea`);
- Campo de validação melhorado (`BetterReCaptcha`).

Para além destes novos campos, foi também desenvolvido um modelo de formulário no ficheiro `templates/fm-form-ipleiria.php` que deve ser copiado para a pasta de modelos da extensão e associado a cada um dos formulários criados para que, em conjunto com os novos tipos de campos desenvolvidos, se garanta que as não conformidades identificadas sejam solucionadas.

4.8.12. Groundwork

Nos últimos anos assistiu-se à proliferação de *frameworks* de suporte à criação de *front-ends* para projetos Web. O Bootstrap[29], Foundation[30], Gumby[31], Pure[32], Ink[33], são apenas algumas das plataformas de suporte que fornecem uma base para a criação de *sites* responsivos com uma biblioteca considerável de funcionalidades que pretendem acelerar o desenvolvimento deste tipo de projetos. Para um projeto desta dimensão é fundamental a adoção de um sistema deste tipo que permita implementar uma plataforma escalável, com uma filosofia e linguagem de comunicação comum.

29 http://getbootstrap.com/

30 http://foundation.zurb.com/

31 http://gumbyframework.com/

32 http://purecss.io/

33 http://ink.sapo.pt/

Numa fase embrionária do desenvolvimento do projeto foram analisadas e testadas uma série de soluções deste tipo no sentido de identificar pontos fortes e fracos de cada um dos sistemas. Inicialmente pensou-se em adotar a *framework* Bootstrap por ter uma comunidade relativamente extensa, dispor de uma boa documentação e pela experiência anterior com o sistema. No entanto, à data dos testes verificou-se que a solução apresentava alguns problemas de acessibilidade na navegação com o teclado e não fornecia informação semântica suficiente a ferramentas de apoio, como leitores de ecrã.

Após vários testes verificou-se que a *framework* GroundworkCSS 2[34] era a solução que mais se adequava aos objetivos definidos para o projeto. Esta *framework* permite criar aplicações Web escaláveis e acessíveis compatíveis com vários formatos de dispositivos, desde dispositivos móveis, *tablets*, computadores, etc. [20] Fornece um sistema para a criação de estruturas Web baseadas em grelha, altamente configuráveis, permitindo um *design* responsivo e adaptativo. Tem uma atenção especial com as questões de acessibilidade, com suporte aos atributos semânticos do WAI-ARIA. É baseado em Sass e na linguagem Coffescript[35], permite prototipagem rápida e integra uma série de ferramentas para otimização de código.

O módulo desenvolvido integra a solução com o WordPress, adicionando suporte pleno às funcionalidades da *framework* tanto na edição de conteúdos ao nível da área administrativa, como na área pública acedida pelos visitantes. Tecnicamente foi necessário implementar algumas alterações de baixo nível no GroundworkCSS, para compatibilizar o sistema com o WordPress e a biblioteca jQuery, bem como introduzir algumas alterações ao funcionamento de algumas funcionalidades (por exemplo, ao funcionamento do menu)[36].

34 https://groundworkcss.github.io/

35 http://coffeescript.org/

36 As alterações implementadas estão publicamente disponíveis no endereço https://github.com/cesperanc/groundwork

Sintaticamente verificou-se que esta *framework* introduz uma linguagem simples e intuitiva para a aplicação de estilos. Por exemplo basta adicionar uma classe CSS com o nome `center` a um elemento para centrar o conteúdo do mesmo; ou que podemos facilmente criar um elemento com duas colunas iguais com o código do Exemplo 4.

```
<div class="row">
   <div class="one half">Coluna 1</div>
   <div class="one half">Coluna 1</div>
</div>
```

Exemplo 4: Exemplo de código HTML compatível com a *framework* GroundworkCSS

O facto de ter sido adotada uma arquitetura modular para este sistema permite que, em caso de necessidade e de forma transparente, se possa substituir este módulo no futuro por outro que integre uma outra qualquer *framework* deste tipo para satisfazer outras necessidades que venham a ser identificadas.

4.8.13. Gallery

Detetou-se já na fase de testes da solução que alguns gestores de conteúdo estavam a criar manualmente galerias de fotografias em notícias, tal como faziam no sistema anterior. Dado que o WordPress disponibiliza uma API própria para este tipo de recursos e que os temas podem utilizar para criar galerias, este módulo fornece os mecanismos para ativação do suporte a esta funcionalidade e definir qual a aparência que estas galerias devem quando são apresentadas aos visitantes. Assim este módulo integra com a API de galerias do WordPress para alterar o código HTML gerado para as galerias, adicionando as classes CSS específicas da *framework* GroundworkCSS para garantir uma aparência consistente e responsiva, bem como para garantir que a estrutura das galerias é reconhecida pelo módulo Colorbox para permitir que o utilizador possa aceder às imagens numa dimensão mais adequada à área disponível, em função do dispositivo de acesso e ao tamanho da própria imagem.

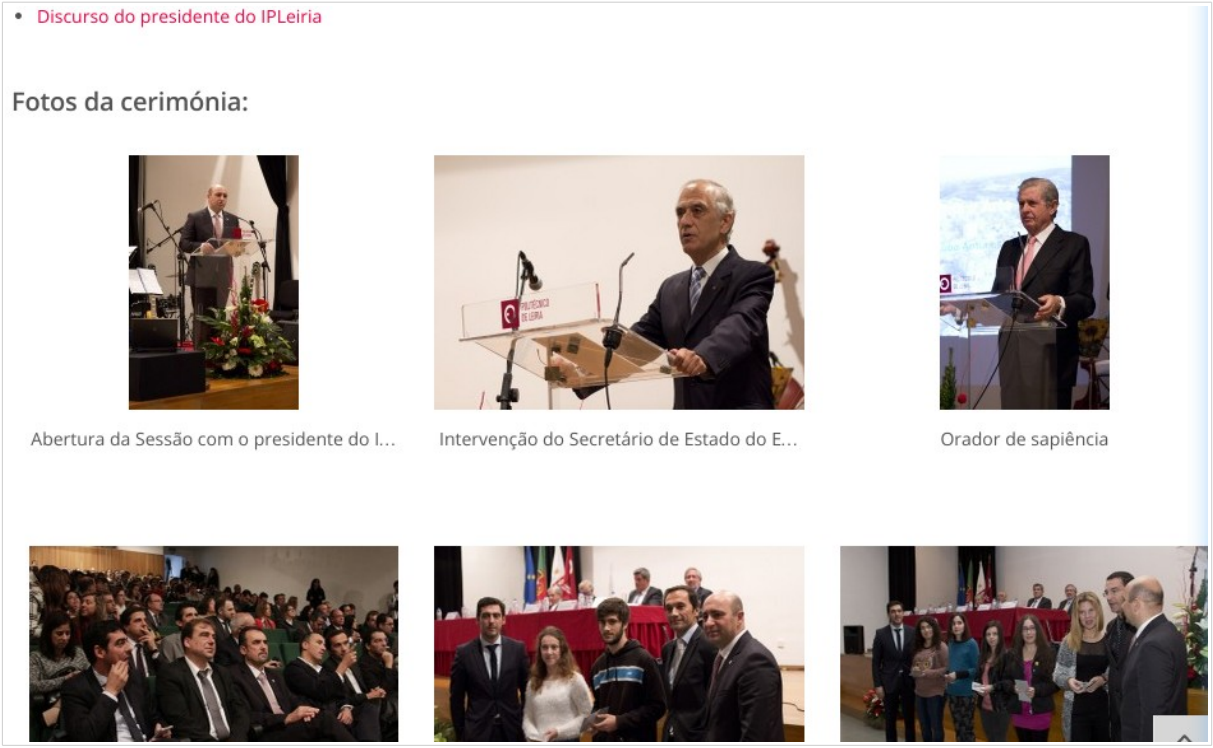


Figura 4.18: Vista pública de uma galeria de imagens criada no WordPress

A inserção e gestão das galerias são feitas com recurso às ferramentas disponibilizadas na área administrativa pelo próprio WordPress e que são comuns a qualquer tema que suporte esta funcionalidade.

4.8.14. Hacks

Hacks foi o nome dado ao módulo que agrega várias correções ou funcionalidades que se considerou que não careciam de um módulo próprio. No âmbito deste projeto este módulo é responsável por:

- Permitir que todos os utilizadores com permissões para publicação de páginas possam inserir HTML não filtrado. Dado que os utilizadores com estas permissões são

funcionários da própria instituição, optou-se por permitir que os utilizadores possam inserir este tipo de conteúdos, como por exemplo código específicos para incorporar vídeos de *sites* externos e que não são nativamente suportados pela API do WordPress. Por omissão o WordPress apenas permite esta funcionalidade a administradores da rede, uma política demasiado restritiva para os objetivos do projeto.

- Adição de uma *tag* `<noscript>` por cada *tag* `<script>` incluída no corpo de uma página. Esta alteração permite tornar as páginas conformes com as WCAG 2.0 que definem esta recomendação, que o WordPress não implementa por omissão.

- Se a extensão Form Manager existir no sistema e estiver ativa, remover o ficheiro JavaScript identificado como `form-manager-js-user` do cabeçalho da página e disponibilizar o mesmo no corpo da página. De forma independente aplica também a mesma alteração com o ficheiro JavaScript da biblioteca jQuery. O objetivo destas alterações é eliminar todos os pedidos de recursos bloqueantes no cabeçalho das páginas HTML geradas, tornando a apresentação de conteúdos mais rápida, segundo as boas práticas vigentes.

- Adicionar informações sobre os tipos de ficheiros com as extensões `cab` e `eps`, para que a sua utilização possa ser autorizada na rede de *sites*. Por omissão o WordPress não permite o envio de ficheiros que não reconhece, mesmo que a sua extensão esteja autorizada. Este módulo descreve o formato destes ficheiros.

- Por fim, este módulo inclui e incorpora as traduções para o ficheiro JavaScript responsável implementar a funcionalidade de voltar ao topo da página.

4.8.15. HistoryBackLink

Nos testes com os utilizadores identificou-se que em determinados contextos de navegação

estes sentiam a falta de um elemento na interface que permitisse voltar à página anterior, principalmente após terem clicando num elemento numa lista de itens, onde perdiam o contexto da página anterior. Este módulo implementa um *shortcode* que permite implementar esta funcionalidade nas páginas onde é inserido, carregando as instruções JavaScript que verifica se existe referência no histórico a uma página anterior, criando dinamicamente uma hiperligação que permite voltar a essa página.

4.8.16. IFrameResizer

O IFrameResizer funciona como um módulo de suporte a outros módulos incluindo uma biblioteca JavaScript que utiliza o método `Window.postMessage()` para permitir a troca de mensagens entre domínios diferentes [21], para contornar restrições de segurança onde ficheiros com instruções JavaScript não podem interagir com elementos de domínios diferentes. Assim esta biblioteca implementa uma API baseada neste método que, com base nas mensagens trocadas, executa as ações solicitadas que sejam suportadas. No contexto deste projeto, isto permite que uma página com um elemento `iframe` colapsado possa solicitar o tamanho do conteúdo interno desse elemento e, com base numa interação do utilizador, expandir esse elemento até que o mesmo tenha o tamanho fornecido pela API.

4.8.17. IPLeiriaBar

Este módulo, com base no conteúdo fornecido por um módulo de um tema específico (e que será abordado numa das secções seguintes), permite criar uma interface para apresentação de uma área dinâmica comum no topo da página. Esta necessidade surge no âmbito das propostas gráficas aprovadas, que previam a criação de uma área comum a todas as páginas e a todos os *sites* do projeto com uma lista de ligações denominada por "Rede IPLeiria". Funcionalmente o utilizador utilizava uma ligação na barra de topo para expandir este componente tendo um acesso facilitado a uma lista de ligações úteis, de uma forma rápida, simples e não intrusiva.

SGCPI

Figura 4.19: Rede IPLeiria

Esta lista de ligações devia ter um ponto de gestão único que permitisse que as alterações efetuadas se propagassem por todos os sistemas que fizessem uso deste elemento. Outro do requisitos estava relacionado com a necessidade de garantir que este componente funcionava não apenas para os *sites* deste projeto, como também para outros projetos Web que quisessem incorporar este elemento gráfico.

Para responder a estes requisitos foi implementado um sistema com um funcionamento duplo que se baseia nos conteúdos definidos para este componente na área administrativa do portal base do projeto. O primeiro modo de operação baseia-se no módulo IFrameResizer para criar um sistema onde um elemento HTML do tipo `iframe` carrega o conteúdo remoto no idioma pretendido, mesmo a partir de domínios base diferentes, sendo o conteúdo apresentado ou ocultado pela interação com um dos elementos visuais presentes na página[37]. Por exemplo, no caso do projeto dos portais do IPLeiria, o conteúdo para este elemento é fornecido através do endereço http://www.ipleiria.pt/IPLeiriaBar.html?lang=en onde o idioma pretendido é definido através valor do parâmetro `lang`. Neste exemplo o código devolvido pelo sistema

37 É fornecido um exemplo de implementação do cliente em HTML e JavaScript no Anexo XXXVI.

contêm todos os estilos necessários e recursos necessários para a correta apresentação do conteúdo dentro do elemento definido, fornecendo a solução ideal para *sites* de terceiros, mesmo que estes utilizem uma *framework* CSS completamente diferente da utilizada neste projeto.

No entanto, apesar deste ser um sistema bastante simples de implementar para *sites* de terceiros, verificou-se que para *sites* locais que sejam parte integrante da solução se podia acelerar consideravelmente a velocidade de carregamento das páginas destes *sites*, eliminado os elementos `iframe` e carregando o conteúdo a pedido. Assim, neste segundo de operação, o conteúdo passa a ser carregado através de um pedido Ajax apenas quando o utilizador interage pela primeira vez com o elemento para apresentação do conteúdo. Por outro lado, o conteúdo carregado é uma versão simplificada do conteúdo devolvido no método anterior apenas com o HTML indispensável para criar o aspeto pretendido, acelerando ainda mais o processo de apresentação da informação reduzindo o *overhead* associado à transferência dos recursos adicionais associados ao método anterior.

O módulo IPLeiriaBar implementa a funcionalidade necessária para fornecer os métodos de operação descritos, fornecendo uma solução tanto para *sites* locais como para *sites* de terceiros, tal como havia sido requerido nas propostas gráficas apresentadas, garantindo a consistência visual mesmo entre projetos distintos da instituição.

4.8.18. IPLeiriaFooter

O IPLeiriaFooter apresenta um modo de funcionamento muito idêntico ao do IPLeiriaBar, permitindo a apresentação de uma zona de rodapé comum a vários *sites* locais e de terceiros. A principal exceção prende-se com a não-existência de um mecanismo de carregamento assíncrono, dado que na proposta gráfica o rodapé institucional está sempre presente e não apresenta nenhum mecanismo de interação que o permita ocultar ou expandir. Assim sendo, para *sites* locais o conteúdo é obtido diretamente a partir da base de dados do sistema,

permitindo assim que esta área seja gerada no mesmo pedido que os outros elementos da estrutura página.

O funcionamento para sites de terceiros é, em tudo semelhante ao módulo IPLeiriaBar, com algumas alterações de sintaxe (Anexo XXXVII).

4.8.19. IPLeiriaTheme

Este módulo serve apenas para associar os estilos básicos do tema base aos temas específicos que se baseiem neste tema.

4.8.20. Language

Um dos requisitos fundamentais deste projeto era o de garantir um sistema que permitisse a gestão e apresentação de conteúdos em vários idiomas. Da análise de soluções pode concluir-se que o Polylang, uma extensão do diretório oficial de extensões do WordPress, era a solução mais adequada para os objetivos definidos. É uma extensão que integra perfeitamente com o interface do WordPress, que fornece uma API extensível e está muito bem classificada num universo de mais de 500 mil instalações.

Este módulo fornece uma camada de abstração da API do Polylang para os restantes módulos dos temas permitindo que, no caso desta extensão deixar de ser a solução ideal para este sistema (por exemplo, por deixar de suportar novas versões do WordPress), a mesma possa ser substituída atualizando apenas o código deste módulo para suportar a nova API sem alterações nos restantes componentes dos temas.

4.8.21. Manage404

Um dos problemas identificados nas reuniões de trabalho entre as várias equipas técnicas

envolvidas no projeto prendia-se com a questão dos endereços e acesso aos recursos da versão anterior do portal. Dado que o processo de migração pressupunha que o novo sistema utilizasse o mesmo endereço base que a plataforma antiga e que não iria existir uma migração direta de conteúdos entre os dois sistemas, colocava-se a questão sobre o que deveria acontecer quando era utilizado um endereço de um recurso válido no sistema anterior que não fosse reconhecido na nova plataforma. A solução passou por desenvolver este módulo, responsável por encaminhar pedidos de recursos não encontrados neste novo sistema para o sistema anterior, temporariamente migrado para um outro endereço. Antes de reencaminhar o cliente para o recurso solicitado na plataforma antiga, o módulo regista a ocorrência nos registos do sistema para que o administrador de sistemas possa mais tarde compilar a origem e o destino destes pedidos, no sentido de solicitar aos administradores de conteúdo dos *sites* de origem que procedam à atualização das ligações antes da instância antiga ser permanentemente desativada.

4.8.22. Menu

Este módulo utiliza a API do WordPress para registar área para a associação do menu principal do *site*. Esta definição será depois utilizada pelo sistema para apresentar o respetivo menu que for associado a esta área principal.

4.8.23. MenuSubWalker

O módulo MenuSubWalker define uma classe PHP, utilizada para processar uma estrutura de menus, modificando os seus parâmetros para que a API do WordPress utilize esta informação para gerar código HTML conforme, com os objetivos definidos para o menu. Concretamente, uma instância desta classe é utilizada para garantir que os elementos de um menu secundário previsto na proposta gráfica assumem o comportamento definido, quando o elemento pai do menu principal ou um dos elementos deste menu secundário está a ser apresentado como

conteúdo principal da página.

4.8.24. MetaInformation

Este módulo é responsável por criar o interface na área administrativa para definição da descrição e palavras-chave do *site*, bem como por adicionar esta informação ao cabeçalho de todas as páginas do mesmo. Esta informação semântica é importante para os motores de pesquisa contextualizarem a temática do *site* e é uma das técnicas fundamentais de SEO.

4.8.25. Options

Options é um módulo secundário responsável por fornecer uma API aos outros módulos para criar páginas, secções e formulários de configuração na área administrativa do WordPress. Com este módulo simplifica-se a implementação de opções de configuração nos restantes módulos evitando assim a duplicação de código.

4.8.26. PostCustomFields

O módulo PostCustomFields é um módulo com uma implementação complexa, mas que adiciona um conjunto de funcionalidades interessantes, principalmente quando combinado com o sistema de modelos fornecido pelo módulo ContentTemplates discutido na secção 4.8.7. Este módulo cria um tipo de conteúdo no WordPress que permite criar e associar campos personalizados de meta-informação a outros tipos de conteúdo no sistema.

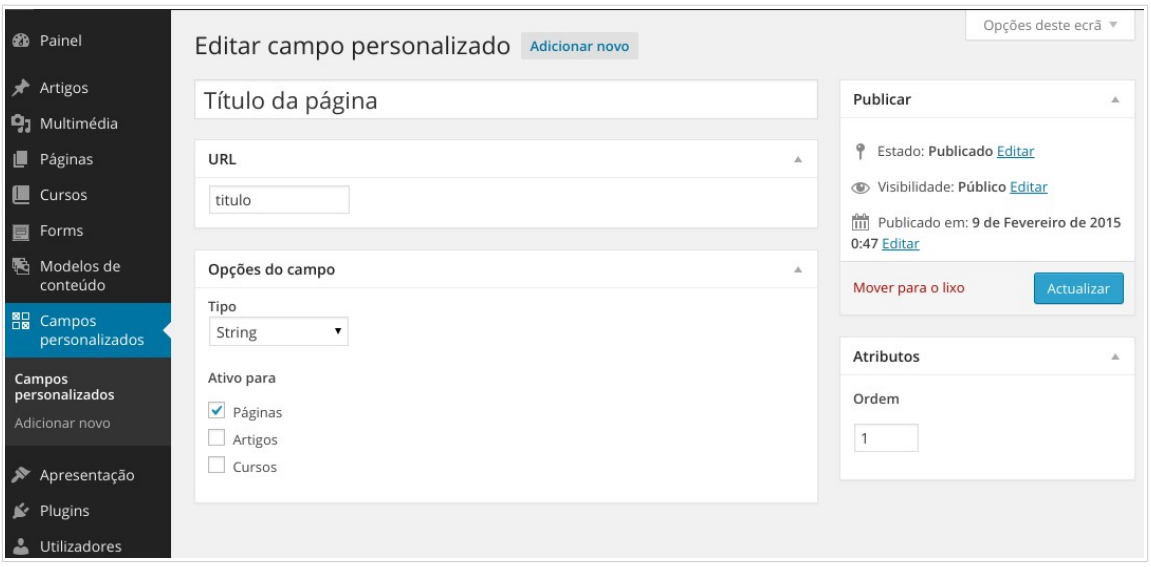


Figura 4.20: Edição de um campo personalizado

No exemplo da Figura 4.20, é definido um campo com o nome `Título da página`, com um identificador/URL `titulo`, do tipo `String`, que está ativo para o tipo de conteúdo `Páginas` e que tem como número de ordem `1`. Com estas definições este módulo cria de forma automática um novo campo com estas caraterísticas no formulário de edição das páginas.

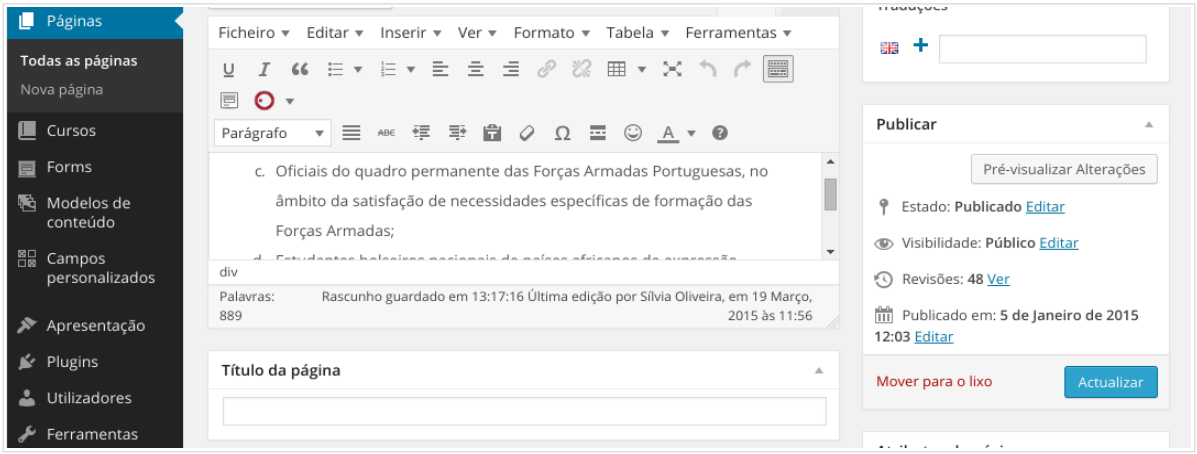


Figura 4.21: Campo personalizado apresentado no formulário de edição

Este campo pode depois ser utilizado num modelo de conteúdo (Figura 4.22), no conteúdo de uma página ou em qualquer outro recurso onde sejam aplicados filtros de conteúdo.

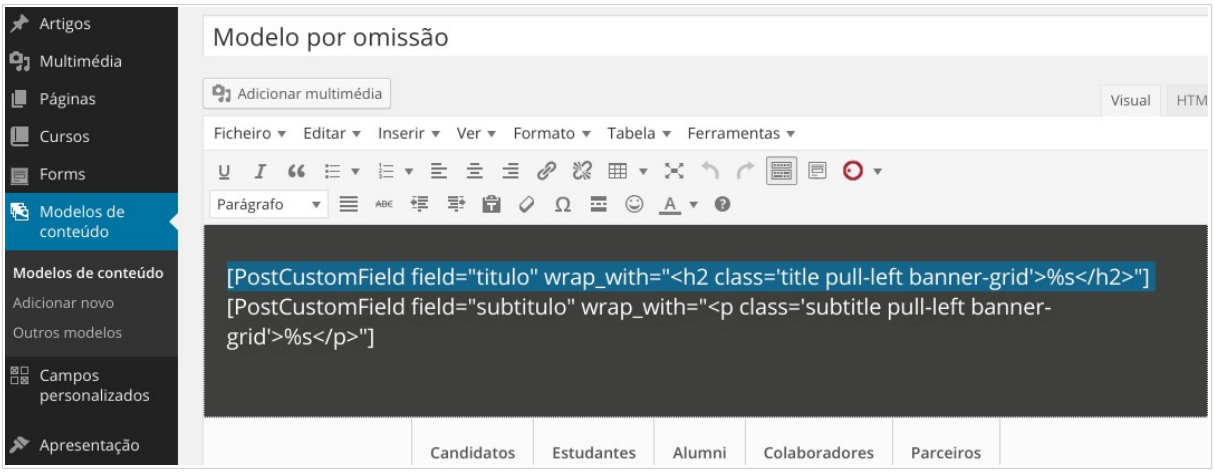


Figura 4.22: *Shortcode* para definição da apresentação de um campo personalizado

Quando a página está a ser renderizada o módulo substitui automaticamente o *shortcode* identificado por `PostCustomField` pelo valor definido no formulário de edição, com a aplicação das opções definidas através do atributos fornecidos.

O *shortcode* suporta os seguintes atributos:

- `field` – atributo obrigatório, com o identificador/URL do campo tal como foi definido no formulário de edição do campo (Figura 4.20).

- `wrap_with` – atributo opcional, com o valor por omissão `%s`. Permite especificar qual o formato que deve ser utilizado para apresentar o conteúdo. Por exemplo, para encapsular o valor definido para o campo com um cabeçalho HTML de nível 2, este parâmetro pode ser definido com o valor `<h2>%s</h2>`, onde `%s` será substituído pelo valor do campo preenchido na área administrativa. Caso o valor do campo não esteja definido ou esteja vazio, nada é devolvido e garantindo que não são apresentados elementos gráficos vazios. Importa ainda referir que carateres como `"` como valor

deste parâmetro podem introduzir comportamentos inesperados por limitações técnicas da própria API do WordPress.

- `args` – atributo opcional que permite enviar dados adicionais às funções de processamento. Em alguns tipos de campos este atributo é utilizado para definir opções de renderização adicionais. O formato de dados é semelhante ao utilizado na codificação e envio de dados não hierárquicos por parâmetros de URL, ou seja o formato `chave1=valor1&chave2=valor2(…)&chaveN=valorN` [22].

Por omissão o módulo suporta campos do tipo texto (`string`) e área de texto (texto, com editor HTML completo). No entanto fornece uma API que pode ser estendida por outros módulos para implementação de tipos de campos adicionais, com recurso a um filtro próprio que permite definir o identificador e etiqueta para o tipo de campo, função ou método a utilizar para renderizar a vista pública do campo e respetivos argumentos, função ou método a utilizar para renderizar a vista de edição com respetivos argumentos, função ou método a utilizar para guardar a informação e filtro de dados a aplicar aos dados a serem guardados.

O módulo disponibiliza ainda internamente acesso a alguns tipos de campos pré-definidos da API do WordPress, com recurso aos seguintes identificadores (a definir como valores do atributo `field` do *shortcode*):

- `post-title` – título definido para o recurso.
- `post-permalink` – endereço público de acesso direto ao recurso.
- `post-excerpt` – excerto/resumo do recurso.
- `post-content` – conteúdo definido para o recurso.
- `post-thumbnail` – imagem associada ao recurso pela funcionalidade de imagem de destaque. Este tipo de campo tem algumas especificidades. Por exemplo, no primeiro *shortcode* do Exemplo 5, com recurso ao atributo `args`, é definido o tamanho da imagem pretendido (através do valor `full` para a variável `thumbnail_size`) sendo também definido o formato pretendido (através da variável `thumbnail_format`), onde

`%1$s` será substituído pelo endereço da imagem a apresentar (sendo ainda possível utilizar os valores `%2$s` e `%3$s` neste formato para representar, respetivamente a largura e altura da imagem) - neste cenário os carateres `<` e `>` são adicionados automaticamente, dado que os mesmos não são suportados como valores de atributos válidos pelo editor WYSIWYG. No segundo *shortcode* do mesmo exemplo, o funcionamento é idêntico, sendo utilizada a variável `format` em substituição da variável `thumbnail_format` que, ao contrário do método anterior, não insere quaisquer carateres adicionais para além dos definidos no formato.

```
[PostCustomField field="post-thumbnail"
args="thumbnail_size=full&thumbnail_format=img src%3D'%1$s'/"]

[PostCustomField field='post-thumbnail'
args='thumbnail_size=full&amp;format=%1$s']
```

Exemplo 5: Exemplos de *shortcodes* para utilização da imagem de destaque associada a um recurso com dois formatos distintos

Este módulo utiliza ainda a API do módulo TinyMCE para adicionar todos os tipos de campos permitidos para um determinado tipo de conteúdo no menu do editor da respetiva área de conteúdo.

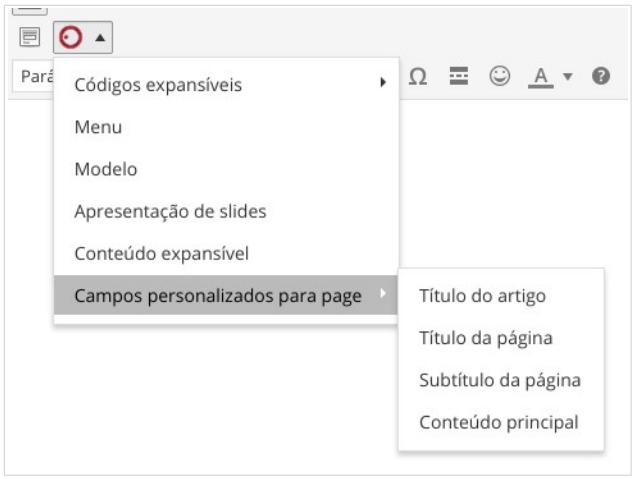


Figura 4.23: Campos personalizados para um conteúdo

4.8.27. PostCustomFieldsFile

Este módulo baseia-se na API do módulo PostCustomFields para criar um tipo de campo que permite a associação de ficheiros a conteúdos, tais como páginas, artigos ou outro tipo de recursos de conteúdo. Assim torna-se possível criar um campo com este tipo, associar o mesmo a um tipo de conteúdo e criar um modelo para esse tipo conteúdo que pressuponha a apresentação de uma ligação para

o ficheiro associado ao conteúdo. O gestor de conteúdos só tem de selecionar o ficheiro a associar ao conteúdo utilizando o interface fornecido e o mesmo será apresentado ao utilizador quando o conteúdo for publicado, tal como definido pelo modelo.

4.8.28. PostTypeContentFilter

O WordPress utiliza taxonomias para categorizar conteúdos que sigam a mesma filosofia de funcionamento dos artigos. Se quisermos, podemos imaginar os artigos como notícias e as taxonomias como categorias às quais podemos associar notícias.

Segundo as propostas gráficas definidas, existem zonas de notícias ou de outros tipos de conteúdo que podem ser colocadas em qualquer secção de uma página. É neste contexto que surge este módulo, como uma solução tecnológica que permite criar este tipo de secções baseada numa instrução de pesquisa, com a aplicação de um modelo pré-desenvolvido disponibilizado na raiz do tema e que será responsável por criar a estrutura visual a apresentar para a secção em causa.

Este módulo utiliza um *shortcode* identificado por `PostTypeFilter`, com suporte para os seguintes parâmetros:

- `query` – atributo obrigatório, com as instruções de pesquisa a utilizar para selecionar os conteúdos a apresentar. Utiliza o formato de parâmetros como os utilizados em endereços Web (`chave1=valor1&chave2=valor2(…)&chaveN=valorN`), cujos valores suportados são os utilizados na classe `WP_Query`[38] da API do WordPress na qual a funcionalidade de pesquisa se baseia para fazer a seleção dos artigos a apresentar. Os resultados da pesquisa são fornecidos ao modelo pela variável `$postTypeContentFilterLoop`, caso o valor do atributo `defer_query` (explicado em abaixo) não seja diferente de `false`.

38 https://codex.wordpress.org/Class_Reference/WP_Query#Parameters

- `template` – sufixo a ser adicionado ao nome `post-type-template-` para localizar o ficheiro PHP na raiz do tema com as instruções para processar os artigos encontrados na pesquisa e gerar o conteúdo HTML a apresentar. O nome do ficheiro do modelo a utilizar deverá seguir o formato `post-type-template-[nome_do_template].php` onde `[nome_do_template]` é o nome específico do modelo e que deve ser fornecido como valor deste parâmetro. Para referência, no Anexo XXXVIII foi disponibilizado o código fonte de um dos modelos utilizados num dos temas desenvolvidos.

- `title` – parâmetro opcional com o título a apresentar. Por omissão este valor não está definido e cabe à implementação do modelo decidir que uso faz deste valor, através da utilização ou não da variável `$postTypeContentFilterTitle`.

- `defer_query` – parâmetro booleano opcional que define se a pesquisa deve ser feita internamente pelo módulo (valor `false`, utilizado por omissão) ou se a mesma será efetuada diretamente no modelo (valor `true`), através da utilização da variável `$postTypeContentFilterQuery` definida pelo módulo e que contem a pesquisa a efetuar. O valor deste parâmetro é fornecido ao modelo pela variável `$postTypeContentFilterQueryDefer`.

```
[PostTypeFilter 
query="post_type=post&category_name=destaques&post__in={sticky_posts}&pos
ts_per_page=9&paged=1" defer_query="true" template="featured"]
```

Exemplo 6: Exemplo de um *shortcode* para o módulo PostTypeContentFilter

O código do Exemplo 6 solicita a inclusão do modelo `featured` (ficheiro `post-type-template-featured.php` na raiz do tema), com a pesquisa a ser executada dentro do próprio modelo e onde a pesquisa deve devolver todos os recursos do tipo `post`, que estejam afetos à categoria com o nome `destaques`, deve ser apresentada uma página de `9` artigos que estejam destacados através da opção fixo. Na área pública este código é traduzido no resultado apresentado na Figura 4.24.

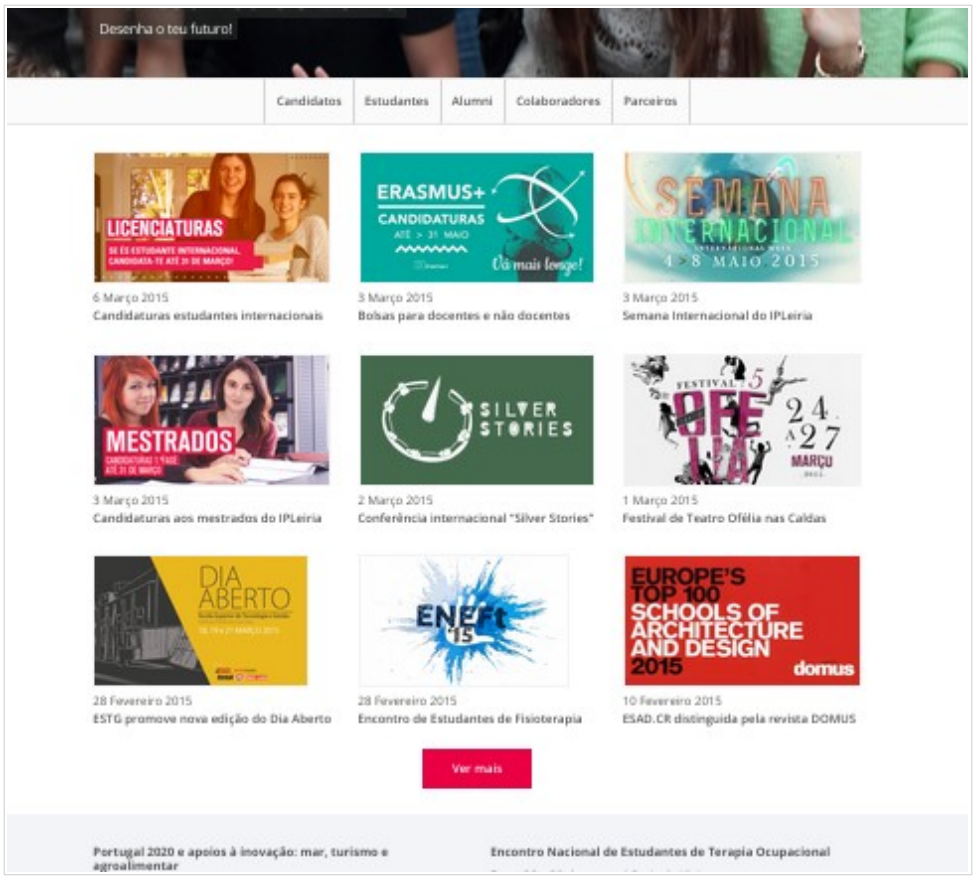


Figura 4.24: Resultado da execução do *shortcode* do Exemplo 6

Este mecanismo flexibiliza imenso alterações como mudança de local da secção, criação de múltiplas secções para categorias diferentes, alterações como aumento ou redução do número de elementos a apresentar, entre muitas outras opções.

Nas iterações mais recentes, a definição do conteúdo dos próprios modelos começou a ser virtualizada para o *backoffice* ao nível de cada um dos elementos a ser apresentado, prevendo-se que será possível ter estes modelos completamente geridos na área administrativa assim que se encontrar uma solução para a implementação de ciclos em modelos, sem comprometer o desempenho ou a segurança do sistema.

4.8.29. PostTaxonomyContentFilter

Este módulo apresenta uma função muito semelhante ao módulo PostTypeContentFilter, mas direcionada à apresentação de taxonomias, ou seja, utiliza modelos para personalizar a apresentação de listas de categorias que pode depois permitir o acesso à lista de artigos a elas associadas.

Este módulo utiliza um *shortcode* identificado por `PostTaxonomyFilter`:

- `taxonomies` – atributo obrigatório, o nome ou identificador das taxonomias a utilizar na pesquisa de categorias.

- `template` – sufixo a ser adicionado ao nome `post-taxonomy-template-` para localizar o ficheiro PHP na raiz do tema com as instruções para gerar o conteúdo HTML a apresentar. O nome do ficheiro do modelo a utilizar deverá seguir o formato `post-taxonomy-template-[nome_do_template].php` onde `[nome_do_template]` é o nome específico do modelo e que deve ser fornecido como valor deste parâmetro. (Exemplo de código fonte para este tipo de modelo no Anexo XXXIX).

- `title` – parâmetro opcional com o título a apresentar. Por omissão este valor não está definido e cabe à implementação do modelo decidir que uso faz deste valor, através da variável `$postTaxonomyContentFilterTitle`.

- `defer_query` – parâmetro booleano opcional que define se a pesquisa deve ser feita internamente pelo módulo (valor `false`, utilizado por omissão) ou se a mesma será efetuada diretamente no próprio modelo (valor `true`), através da utilização das variáveis `$postTaxonomyContentFilterTaxonomies` e `$postTaxonomyContentFilterAttributes` definidas pelo módulo e que contem a pesquisa a efetuar. O valor deste parâmetro é fornecido ao modelo pela variável `$postTaxonomyContentFilterQueryDefer`.

Para além destes parâmetros específicos do módulo, são ainda suportados todos os argumentos utilizados pela função `get_terms`[39] da API do WordPress como atributos do *shortcode* `PostTaxonomyFilter` e que são utilizados para refinar ainda mais os resultados da pesquisa. Por questões de suporte da *shortcode* API do WordPress, todos os argumentos da função definidos como vetores, devem ser enviados como uma lista de elementos separados com o carater `,`.

```
[PostTaxonomyFilter taxonomies="type" template="courses-types"
no_wrapper="true" include="110,116,269"]
```

Exemplo 7: Exemplo de um *shortcode* para o módulo PostTaxonomyContentFilter

4.8.30. RemoteMenu

Um dos requisitos do projeto dita que deve ser mantida a consistência gráfica entre os vários *sites* do sistema tornando a transição entre os mesmos completamente transparente e fluida para o utilizador. Como os temas específicos baseiam a sua funcionalidade no tema base (responsável para definição da linha gráfica base), consegue-se garantir uma consistência visual relativamente ao aspeto de todos os *sites*. No entanto, como existe isolamento de funcionalidades e de conteúdos através da separação lógica dos vários *sites* no sistema, garantir a consistência entre menus na transição entre os vários *sites* da solução torna-se uma questão algo complicada, dado que o WordPress não permite troca de dados entre *sites* distintos mesmo que estejam sob uma mesma rede ou sistema, incluindo informações sobre menus.

Este módulo implementa uma solução tecnológica que visa permitir o acesso aos menus dos vários *sites* existentes na rede, permitindo que os administradores possam adicionar aos menus dos seus *sites* itens de menu disponíveis em menus de outros *sites* da mesma rede.

39 http://codex.wordpress.org/Function_Reference/get_terms#Possible_Arguments

SGCPI

O módulo trata de fazer a gestão automática dos itens partilhados, garantindo que se os itens individuais ou grupos de itens (sub-menus) forem atualizados, estas alterações são refletidas em todos os *sites* aos quais estão associados. Assim, se no *site* de origem de um determinado sub-menu for adicionado ou removido um item, esta alteração é automaticamente refletida em todos os menus dos outros *sites* que referenciam esse mesmo sub-menu. A extensão implementa ainda mecanismos de gestão da *cache* partilhada entre os vários *sites*, no sentido de maximizar o desempenho da solução.

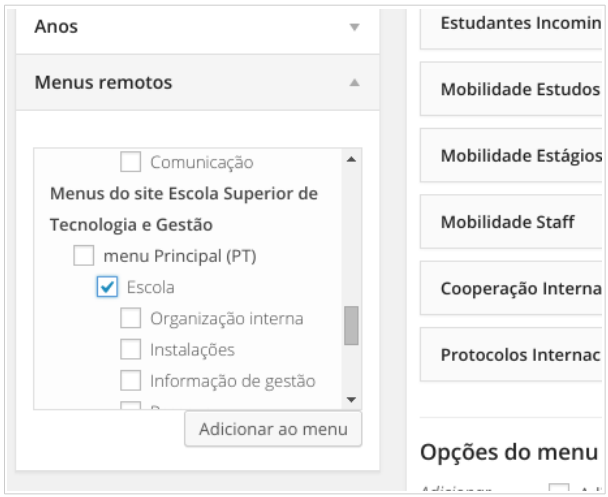


Figura 4.25: Componente de seleção de menus remotos a adicionar ao menu

Para o utilizador o funcionamento da solução é completamente transparente, dando a sensação que se trata de apenas um portal único.

4.8.31. ResponsiveImages

Com a disseminação em larga escala da computação móvel, é cada vez maior o número de acessos a *sites* com dispositivos móveis. Num mercado bastante heterogéneo onde, equipamentos de várias marcas que procuram fatores de diferenciação para cativar o maior número de clientes em relação aos seus concorrentes, existem várias centenas de equipamentos móveis com diferentes funcionalidades e caraterísticas, tais como poder de processamento, tamanho e resolução do ecrã, velocidade de acesso à rede, etc. Por outro lado, os pacotes de dados móveis para acesso à Internet ainda têm um custo relativamente elevado (quando comparados com os acessos pela rede física), muitas vezes com restrições e limites

de tráfego, onde velocidade (ainda) é mais reduzida. Isto traz alguns desafios onde, por um lado temos ecrãs com um número de pixeis por polegada cada vez mais elevado, que exigem imagens com mais qualidade e com um tamanho maior; por outro temos limites no volume de dados que podemos transferir. Encontrar um ponto de equilíbrio entre estes dois universos aparentemente opostos não é um problema com uma resolução simples.

Este módulo introduz o suporte à *tag* `picture`[40] do HTML que permite a definição de múltiplas fontes para uma mesma imagem e que neste momento já é suportada por alguns dos navegadores Web mais recentes. Quando um gestor de conteúdo envia uma imagem para o sistema, este módulo fornece indicações à API do WordPress para criar várias variações dessa imagem com diferentes tamanhos. Quando o conteúdo é publicado e consultado por um utilizador, o módulo analisa todas as imagens referenciadas pelo conteúdo do documento e constrói um elemento `picture` para cada uma delas com indicação de todas as variações de tamanho e respetiva localização, informações que um navegador pode utilizar para selecionar a versão da imagem com o tamanho mais adequado em função da resolução do ecrã. Isto permite reduzir o volume de dados transferidos e o tempo necessário para apresentar a página.

O funcionamento do sistema é totalmente transparente para o utilizador e para o gestor de conteúdos. O gestor de conteúdos até pode enviar uma imagem com uma resolução 4K, mas se o navegador do utilizador apenas suportar a resolução 720p, será apenas esta imagem que será transferida pelo equipamento do utilizador.

Para os navegadores que ainda não compatíveis com esta especificação do HTML, o módulo incorpora ainda suporte para a biblioteca JavaScript Picturefill[41] para garantir o suporte parcial a estes clientes mais desatualizados.

40 https://html.spec.whatwg.org/multipage/embedded-content.html#the-picture-element

41 http://scottjehl.github.io/picturefill/

4.8.32. Robots

Este módulo é responsável por, com base no conhecimento que tem sobre o sistema e do modo de organização da informação, criar uma versão virtual do ficheiro `robots.txt` com indicações sobre como a informação gerada pelo sistema deve ser indexada pelos *crawlers* dos motores de pesquisa.

```
User-agent: *
Disallow: /wp-admin/
Disallow: /wp-includes/
Disallow: /wp-content/plugins/
Disallow: /wp-content/themes/ipleiria/
Allow: /
Sitemap: http://www.ipleiria.pt/sitemap.xml
```

Exemplo 8: Exemplo do conteúdo do ficheiro `robots.txt` gerado pelo módulo Robots

4.8.33. SiteMapXml

Como complemento à informação fornecida pelo módulo Robots, o módulo SiteMapXml gera um ficheiro `sitemap.xml` virtual com uma estrutura XML normalizada com a apresentação das ligações para acesso aos conteúdos do *site* (e respetivas prioridades de indexação) aos motores de pesquisa, no sentido de otimizar e facilitar o processo de indexação de conteúdos disponibilizados no sistema.

O módulo associa também uma folha de estilos XSL para transformar o documento XML gerado num formato visual de leitura mais simples para os humanos (Figura 4.26). Define ainda cabeçalhos HTTP para gestão da *cache* e meta-dados adicionais para o ficheiro virtual.

Mapa XML do site Politécnico de Leiria

Este é um mapa XML do site para ajudar os motores de pesquisa.

#	Endereço	Prioridade
1	http://www.ipleiria.pt/	100%
2	http://www.ipleiria.pt/cursos/	90%
3	http://www.ipleiria.pt/sas/inicio/	90%
4	http://www.ipleiria.pt/internacional/	90%

Figura 4.26: Mapa do *site* no formato XML

4.8.34. SiteMap

O módulo SiteMap fornece uma URL própria para a apresentação do mapa do *site* construído a partir da análise da informação obtida a partir da análise ao menu principal do *site*. Nesta primeira fase do projeto, os principais destinatários desta funcionalidade são os utilizadores que recorrem aos leitores de ecrã e/ou navegação por teclado, para que estes tenham um acesso facilitado à estrutura principal do portal.

4.8.35. SiteStructureInformation

As pessoas com deficiência visual utilizam ferramentas como leitores de ecrã para interpretarem a estrutura de uma página e perceberem como é que a informação está organizada. No entanto, em portais com uma quantidade significativa de informação, este processo pode ser complexo e nem sempre é simples identificar os elementos que podem ser utilizados como pontos de referência para a navegação. Este módulo cria uma área de conteúdo que os administradores do sistema podem utilizar para descrever genericamente a estrutura do *site* que funciona como um pequeno guia de apoio à navegação. Se o guia estiver disponível, o acesso ao mesmo pode ser feito na área pública do *site* com recurso à navegação com o teclado, sendo uma das primeiras hiperligações das páginas.

4.8.36. SideBars

Este módulo foi o resultado da implementação de uma prova de conceito que acabou por não ser necessária no âmbito da implementação deste sistema. Este módulo seria responsável por criar e gerir as áreas para disponibilização de *widgets* do WordPress em vários idiomas (com recurso à API do módulo Language), bem como por fornecer uma API para a apresentação das mesmas. Apesar do módulo estar completamente funcional, o mesmo foi desativado por não ser necessário nesta fase do projeto.

4.8.37. Utilitities

O módulo Utilitities disponibiliza uma biblioteca de funções auxiliares comuns para os temas e restantes módulos, utilizada ao longo da solução para processar filtros de datas, análise de pedidos, etc.

4.8.38. ViewportChecker

Este módulo adiciona suporte a uma biblioteca JavaScript que permite adicionar na área pública, classes CSS a determinados elementos HTML relativamente à sua posição quanto à área de conteúdo e ao deslocamento da página no navegador Web. Isto permite, por exemplo, criar animações quando o deslocamento da página atinge uma posição próxima de um determinado elemento. Para o efeito deve ser adicionada a classe CSS `animated` ao elemento se pretende animar e definir os seguintes atributos:

- `data-animation` – atributo obrigatório e que define qual a classe CSS que deve ser adicionada ao elemento quando é atingido o ponto pretendido.
- `data-animation-offset` – atributo opcional (por omissão, `100`), que define o

número de pixeis antes ou depois do elemento onde a classe CSS deve ser adicionada ao elemento em causa.

- `data-animation-repeat` – atributo opcional (por omissão, `false`), que define se a animação deve ser repetida caso a posição de animação seja novamente atingida.

- `data-animation-callback` – atributo opcional, com uma função JavaScript com a assinatura `function(elemento, acao)` que será executada após a adição ou remoção da classe especificada ao elemento.

4.8.39. WpCleanup

O módulo WpCleanup utiliza a API do WordPress para personalizar algumas funcionalidades da plataforma, tais como remoção do suporte a comentários, remoção de meta-informações não utilizadas, personalização do formato do excerto, entre outras alterações consideradas necessárias.

4.8.40. TechnicalSheet

Este módulo disponibiliza uma área ao administrador do *site* onde o mesmo poderá criar a ficha técnica do sistema, para descrever a solução tecnológica e apresentar a equipa que esteve envolvida no projeto. Se preenchida, esta página de informação é acessível clicando na ligação associada à mensagem de direitos de autor.

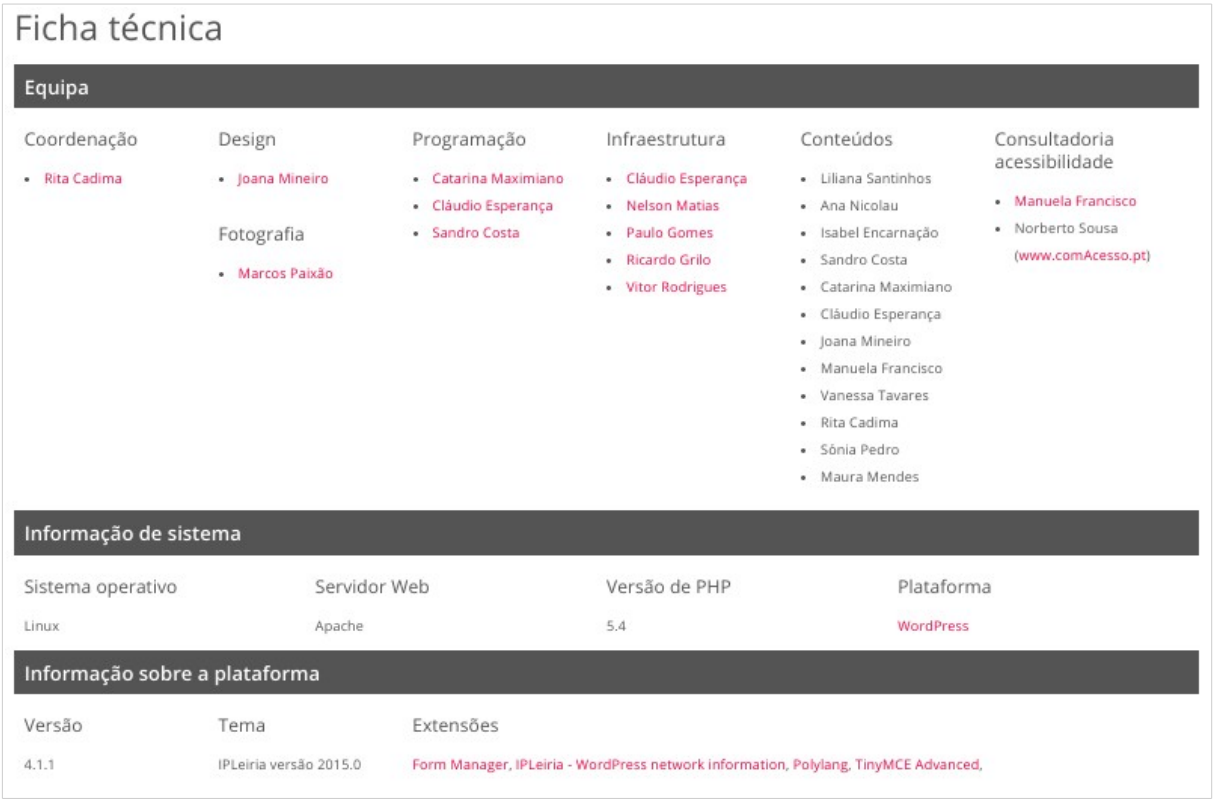


Figura 4.27: Página da ficha técnica de um dos *sites*

O conteúdo desta ficha pode ser definido manualmente ou podem ser utilizados os *shortcodes* disponibilizados pelo módulo para a obtenção de algumas informações do sistema, a saber:

- `copyright_information` – imprime a mensagem de direitos de autor. Suporta o atributo opcional numérico `year` para definição do ano inicial do projeto e `copyright` com o texto da mensagem dos proprietários dos direitos de autor.

- `site_technical_sheet_theme_info` – imprime uma informação a partir dos meta-dados do tema (ficheiro `style.css`), tais como nome, descrição, autor, etc. Suporta o atributo obrigatório `retrieve_tag` com o nome da *tag* a obter, o atributo opcional

`wrap_with` com a estrutura HTML onde o resultado deve ser encapsulado, o atributo opcional `split_by` com o(s) carater(es) a utilizar para dividir os elementos em múltiplas linhas e o atributo opcional `row_format` com o formato utilizar para cada um dos elementos obtidos.

- `site_technical_sheet_authors` – imprime uma lista dos autores dos conteúdos do *site* atual. Suporta o atributo opcional `link_format` com formato da ligação a utilizar para cada autor onde, `%1$s` corresponde ao endereço definido no perfil do autor, `%2$s` um título genérico para associar à hiperligação e `%3$s` é o nome do autor. Com exceção do atributo `retrieve_tag`, suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`.

- `site_technical_sheet_system_os` – apresenta informações sobre o sistema operativo que suporta a solução. Com exceção do atributo `retrieve_tag`, suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`.

- `site_technical_sheet_system_server` – apresenta informações sobre o serviço HTTP que suporta a solução. Com exceção do atributo `retrieve_tag`, suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`.

- `site_technical_sheet_system_php` – apresenta informações sobre a versão de PHP da solução. Com exceção do atributo `retrieve_tag`, suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`. Adicionalmente suporta o parâmetro `version_format` com formato numérico a apresentar para a versão, onde `%1$d` corresponde ao código da versão principal do PHP, `%2$d` ao código da versão intermédia e `%3$d` ao código da versão de lançamento.

- `site_technical_sheet_system_platform_version` – apresenta informações sobre a versão de WordPress que suporta a solução. Com exceção do atributo `retrieve_tag`,

suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`.

- `site_technical_sheet_system_platform_theme` – apresenta informações sobre o tema em uso. Com exceção do atributo `retrieve_tag`, suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`. Adicionalmente suporta o parâmetro `version_format` com formato a apresentar, onde `%1$s` corresponde ao nome do tema e `%2$s` à respetiva versão.

- `site_technical_sheet_system_platform_plugins` – imprime a lista de extensões ativas no tema. Suporta o atributo opcional `link_format` com formato da ligação a utilizar para cada extensão onde, `%1$s` corresponde ao endereço da extensão ou do autor da extensão, `%2$s` a um título genérico a associar à hiperligação e `%3$s` com o nome da extensão. Com exceção do atributo `retrieve_tag`, suporta ainda todos os parâmetros descritos para o *shortcode* `site_technical_sheet_theme_info`.

No Anexo XL apresenta-se um exemplo do conteúdo HTML completo de uma ficha técnica com recurso aos *shortcodes* descritos.

4.9. Temas e módulos específicos desenvolvidos

Na secção anterior foi apresentado o tema base da solução com os módulos que fornecem as funcionalidades fundamentais herdadas por todos os temas específicos baseados neste tema. Apesar de ser um tema completamente funcional, nesta arquitetura definiu-se que este tema não seria diretamente utilizado em nenhum projeto, de modo a afastar a tentação de serem introduzidas funcionalidades específicas num tema que se quer genérico e de âmbito mais geral. Todos os projetos que contenham especificidades relativamente ao tema base, devem implementá-las em módulos específicos encapsulados em temas próprios exclusivos do respetivo projeto, permitindo que os mesmos possam evoluir de forma isolada ao longo do

tempo. Se existirem módulos comuns entre dois ou mais temas, estes devem ser duplicados na pasta `thememodules` específica de cada um, não contaminando o tema base com funcionalidades específicas para apenas alguns dos *sites* da solução.

Nesta secção serão apresentados os principais temas específicos desenvolvidos, bem como respetivos módulos e funcionalidades.

4.9.1. Tema `ipleiria_basic`

O tema `ipleiria_basic` é o tema por omissão para novos *sites* de projetos que não necessitem de funcionalidades específicas relativamente ao tema base. Neste projeto foi utilizado em *sites* como a Direção de Serviços de Informática[42] e o Provedor do Estudante[43], e incorpora alguns módulos e modelos específicos:

Módulo `CustomLogo`

Este módulo recorre à API do WordPress para ativar a funcionalidade necessária para a definição de logótipos personalizados no cabeçalho de todas as páginas do tema, permitindo que este tema possa ser aplicado em páginas de unidades e projetos com logótipos próprios. Por omissão é apresentado o logótipo oficial da instituição.

Módulo `Posts`

O módulo `Posts` tem única finalidade configuração da solução para indicar que a mesma suporta a API do WordPress para a associação de imagens de destaque a recursos, permitindo que um gestor de conteúdos possa, por exemplo, definir uma imagem de destaque para um artigo.

42 http://www.ipleiria.pt/dsi

43 http://www.ipleiria.pt/provedordoestudante

SGCPI

Módulo `PostsLink`

Por omissão as notícias devem ter sempre conteúdo associado mas, por vezes, existe a necessidade de criar notícias que não são mais do que uma referência a uma notícia ou destaque de um outro *site* e para as quais não é prático estar a criar um conteúdo redundante. É neste contexto que surge este módulo, como um sistema que adiciona um campo ao formulário de edição dos artigos, permitindo que um gestor conteúdos possa associar uma hiperligação a esses recursos que, quando preenchida, modifica o comportamento do WordPress fazendo com que ao ser ativada, a hiperligação de acesso ao conteúdo na área pública encaminhe o utilizador para a ligação definida neste campo em vez de encaminhar o utilizador para o conteúdo local da notícia.

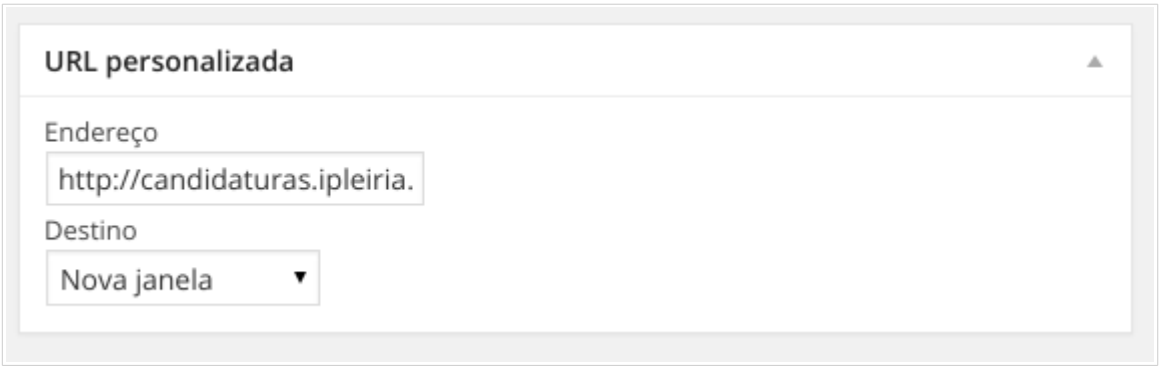


Figura 4.28: Campo de definição de um endereço personalizado para um artigo

Modelo específico `post-type-template-posts`

Define um modelo para o módulo PostTypeContentFilter apresentado na página 91, que define uma estrutura específica para apresentação de notícias em zonas de conteúdo de páginas dos *sites* desenvolvidos.

4.9.2. Tema `ipleiria_courses`

A educação e a formação são o *core business* do IPLeiria, uma área estratégica para a qual a

divulgação de informação sobre a oferta formativa assume um papel fundamental. Neste sentido é essencial a existência de uma solução tecnológica que permita satisfazer as necessidades específicas desta área de negócio, onde a usabilidade e a forma como a informação é apresentada ao utilizador é de extrema importância para cativar os visitantes a se tornarem futuros alunos da instituição.

Este tema encapsula as funcionalidades específicas para a disponibilização de um sistema para gestão e apresentação da oferta formativa do IPLeiria.

Módulo `Courses`

Este é, possivelmente, um dos módulos mais complexos de toda a solução. Encapsula em si grande parte das funcionalidades para gestão da oferta formativa, tais como, gestão de cursos, gestão de áreas de estudo, gestão de idiomas dos cursos, gestão de regimes, gestão de escolas, gestão de tipos de formação e gestão dos campos de dados associados a cada curso, incluindo o plano curricular.

A sua funcionalidade está, também ela, organizada por módulos que tratam partes específicas do problema. Estes módulos estão divididos em ficheiros, cujo conteúdo que passamos a descrever:

- `css/coursesCurricularPlan(.dev).css` – contém os estilos para a personalização do aspeto do campo de edição do plano curricular na área administrativa.

- `js/CourseCurricularPlan(.dev).js` – contém as instruções JavaScript que definem o comportamento específico dos elementos que compõem o plano curricular na área pública do *site*.

SGCPI

- `js/CoursesCurricularPlanAdmin(.dev).js` – define o comportamento dos elementos do campo de edição do plano curricular associado aos cursos na área administrativa do *site*. São instruções JavaScript baseadas na API do Underscore[44] (para criação de HTML a partir de modelos específicos, com JavaScript e dados no formato JSON), do Backbone[45] (para sincronização de dados com base no modelo MVP, uma derivação do MVC, utilizado para construção de interfaces gráficos), do TinyMCE (para incorporação do editor gráfico num dos campos) e do WordPress (para funções de suporte).

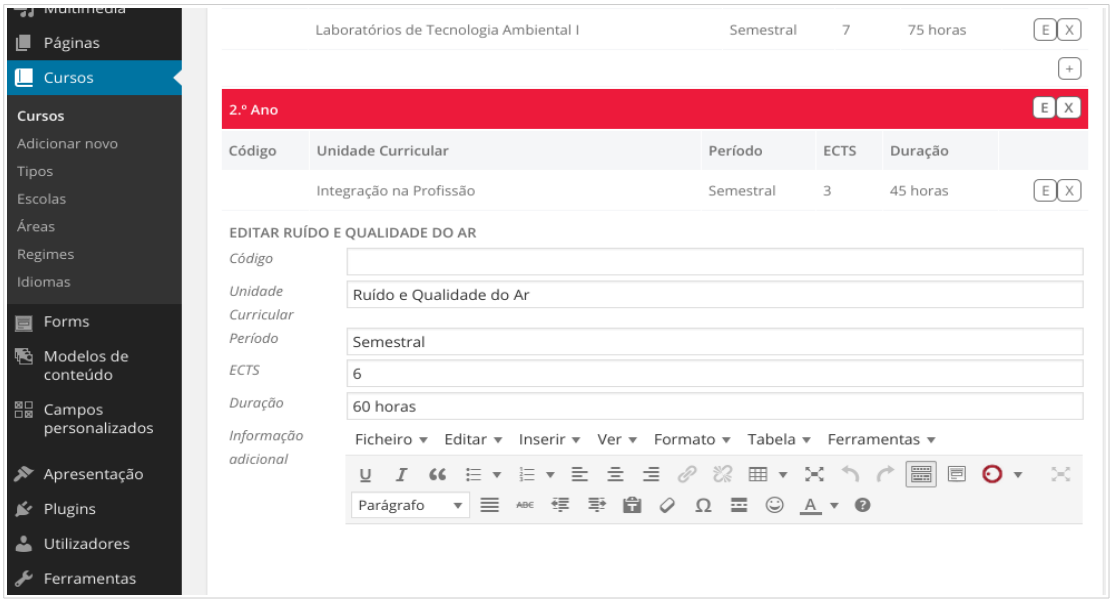


Figura 4.29: Interface de edição do plano curricular de um curso

- `js/CoursesCategoryTaxonomiesListBox(.dev).js` – contém as instruções JavaScript que definem o comportamento específico dos componentes para a filtragem de cursos na lista de cursos.

44 http://underscorejs.org/

45 http://backbonejs.org/

- `traits/CoursesCategoryTaxonomy.php` – fornece uma API que permite implementar as funcionalidades necessárias para a criação de novos tipos para categorização de cursos (taxonomias). Basicamente permite criar tipos de filtros adicionais para além dos tipos, áreas, escolas, regimes e idiomas, de um modo simples e rápido.

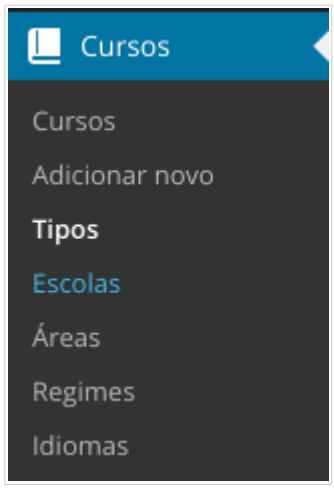


Figura 4.30: Taxonomias de cursos disponibilizadas

- `traits/CoursesCategoryPostColumns.php` – fornece uma API para adicionar colunas adicionais à lista de cursos na área administrativa, permitindo identificar de uma forma mais visual e imediata quais as categorias que estão a ser aplicadas a um determinado curso.

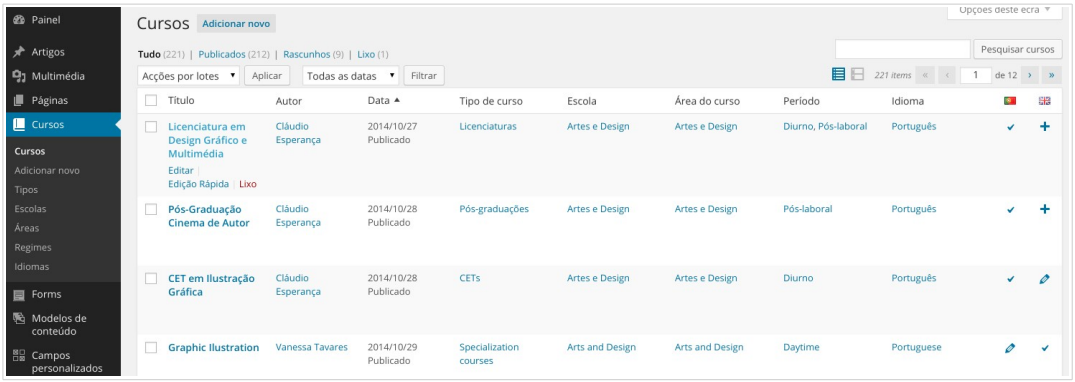


Figura 4.31: Lista de cursos com a apresentação das colunas adicionais

- `traits/CoursesCategoryThumbnail.php` – fornece uma API que permite associar, a partir da Biblioteca do WordPress, uma imagem personalizada a uma determinada categoria para que a mesma possa ser utilizada como um destaque visual na área pública do *site*.

- `traits/CoursesCategoryExtraFormFields.php` – fornece uma API para adicionar

campos adicionais (para além do título e descrição) ao formulário de edição de categorias, permitindo utilizar estas informações em outras secções do *site*.

- `CoursesCurricularPlan.php` – com recurso à API do módulo PostCustomFields (descrito na página 86), define um tipo de campo personalizado adicional com a estrutura de um plano curricular. Com este tipo de campo, um administrador pode criar um novo campo personalizado deste tipo que associa ao tipo de conteúdo `Curso`, disponibilizando um interface que o gestor de conteúdos poderá utilizar para preencher o plano curricular de cada curso.

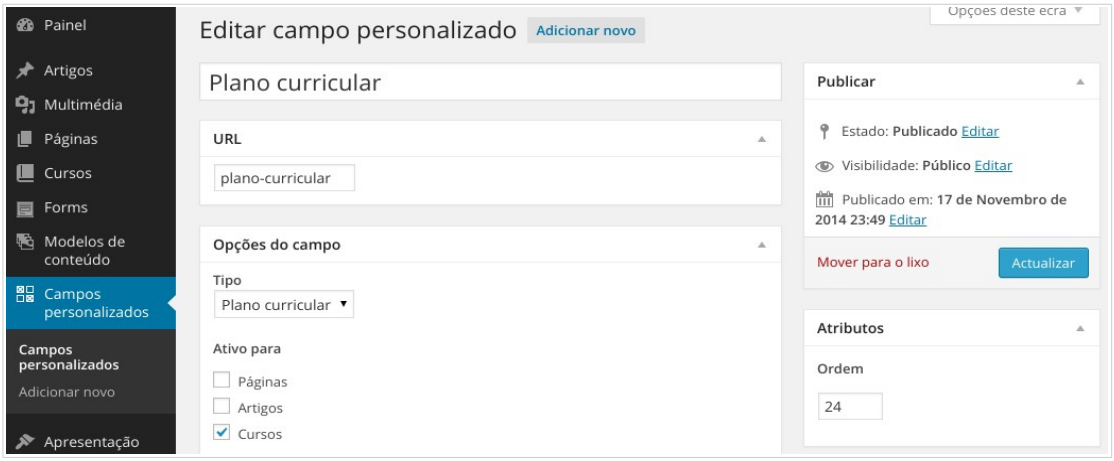


Figura 4.32: Campo personalizado para o plano curricular de um curso

- `CoursesCategoryType.php` – com recurso aos sub-módulos anteriormente descritos, cria uma categoria específica para o tipo de oferta formativa, com suporte a uma imagem de destaque por cada tipo de curso, suporte a uma descrição HTML e cujo valor do tipo deve ser apresentado nas colunas da lista de cursos na área administrativa.

- `CoursesCategorySchool.php` – baseia-se nos módulos anteriores para registar uma categoria para associar cursos a escolas, com uma descrição HTML e um acrónimo para a escola, bem como a apresentação o nome da escola nas colunas da lista de

cursos na área administrativa.

- `CoursesCategoryPeriod.php` – baseia-se nos módulos anteriores para registar uma categoria para associar cursos a regimes, com uma descrição HTML e com a apresentação do valor do regime nas colunas da lista de cursos na área administrativa.

- `CoursesCategoryLanguage.php` – baseia-se nos módulos anteriores para registar uma categoria para associar cursos a idiomas, com uma descrição HTML e com a apresentação do valor do idioma nas colunas da lista de cursos na área administrativa.

- `CoursesCategoryArea.php` – baseia-se nos módulos anteriores para registar uma categoria para associar cursos a áreas do conhecimento, com uma descrição HTML, uma imagem de destaque e com a apresentação do valor especificado para a área nas colunas da lista de cursos na área administrativa.

- `Courses.load.php` – núcleo do módulo responsável por carregar cada um dos módulos dos cursos bem como parametrizar o funcionamento da lógica de negócio associada à gestão da oferta formativa da instituição. As principais funcionalidades deste módulo são as seguintes:

    - Registo do tipo de conteúdo específico para a gestão de cursos no sistema.

    - Registo dos ficheiros JavaScript necessários para a execução da funcionalidade necessária para a apresentação dos cursos na área pública do *site*.

    - Definição das expressões regulares para a regras dos endereços associadas à apresentação e filtragem de cursos com base no endereço fornecido. Por exemplo, este módulo deve transformar um endereço do tipo http://www.ipleiria.pt/cursos/course/school/**estg**/type/**mestrado**/ numa lista de cursos do tipo `mestrado` e afetos à escola `estg`. Esta funcionalidade é

implementada com a definição de regras específicas com recurso à API do WordPress.

- Configuração do suporte para multi-idioma para este tipo de conteúdo.

- Carregamento dos modelos específicos da solução para as páginas de filtragem de cursos.

- Personalização dos endereços associados às categorias com base nas regras de endereços definidas.

- Virtualização dos modelos associados à apresentação da lista de cursos, elementos de uma lista de cursos e interface de apresentação de um curso de forma isolada. No Anexo XLI, Anexo XLII e Anexo XLIII são apresentados exemplos de cada um destes modelos, onde se recorre à API fornecida por módulos como o PostCustomFields, o ContentTemplates, o ActionExecutionContentFilter, o PostTypeContentFilter e o PostTaxonomyContentFilter para a construção da interface gráfica das várias áreas de apresentação de informação associada aos cursos. Estes modelos recorrem aos vários campos personalizados previamente criados no sistema e associados aos cursos que, após serem preenchidos pelo gestor de conteúdos, são apresentados ao visitante nos locais indicados nestes modelos. Isto permite que, tal como aconteceu ao longo do ciclo de desenvolvimento do projeto, fossem sendo adicionados, modificados e removidos campos, bem modificada a estrutura de apresentação da informação, de acordo com o refinamento das necessidades e ajuste dos objetivos deste sub-projeto.

Modelo específico `post-taxonomy-template-courses-types`

Define um modelo para o módulo PostTaxonomyContentFilter apresentado na página 94, para personalizar a estrutura HTML para a apresentação dos vários tipos de cursos existentes no sistema. Isto permite definir como será apresentado o título, a descrição e a imagem de destaque associada a cada um dos tipos de cursos.

Modelo específico `post-course`

Processa os conteúdos a apresentar na página de filtragem de cursos com base nos modelos virtuais definidos na área administrativa pelo administrador do sistema.

Modelo específico `post-type-template-courses`

Define um modelo para o módulo PostTypeContentFilter apresentado na página 91, para a definição da estrutura de apresentação de cursos nas áreas de conteúdos de páginas, caso isto seja necessário.

Módulo `CoursesSearch`

À data da elaboração deste relatório, estavam registados no sistema cerca de 220 cursos. Apesar dos filtros fornecerem uma funcionalidade que permite refinar a informação e encontrar um curso com alguma facilidade, um dos requisitos definidos na proposta gráfica aprovada para a página inicial do *site* de cursos ditava que devia ser apresentado um campo de pesquisa que permitisse pesquisar por cursos de forma simples e rápida. Este módulo fornece o interface e a lógica necessária para implementar esta funcionalidade, onde um visitante pode escrever na caixa de pesquisa da página inicial dos cursos o nome do curso, área, escola, tipo, regime ou idioma em que está interessado e o sistema irá fazer uma pesquisa pelo nome

desses recursos, apresentando a informação que o sistema considerar mais relevante. O algoritmo de pesquisa é relativamente simples, embora esteja previsto o desenvolvimento de um algoritmo mais abrangente numa futura iteração, adicionando suporte a pesquisas um pouco mais avançadas.

A pesquisa é feita de forma assíncrona ao sistema como recurso a JavaScript. Foi tida em consideração a associação de instruções semânticas para tornar o componente mais acessível a tecnologias de apoio.

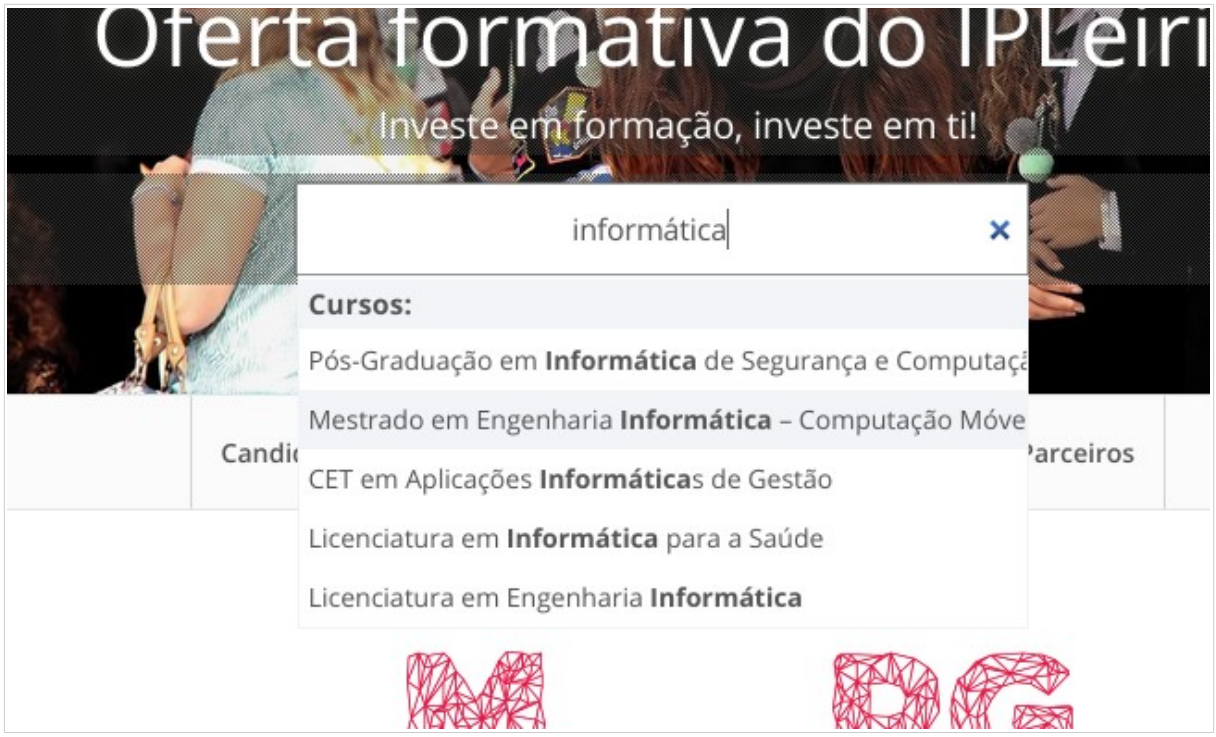


Figura 4.33: Exemplo da utilização da caixa de pesquisa de cursos

4.9.3. Tema `ipleiria_ipleiria`

Este é o tema do portal principal da solução[46] e contém um conjunto de módulos e temas específicos para a concretização da funcionalidades previstas para este *site*:

46 http://www.ipleiria.pt/

Módulos `IPLeiriaCustomBar` e `IPLeiriaCustomFooter`

Estes módulos são responsáveis por criar o interface de administração e por definir os valores a serem apresentados, respetivamente, pelos módulos IPLeiriaBar e IPLeiriaFooter referidos anteriormente. A definição destes valores neste tema na raiz da solução, permite garantir que o cabeçalho e rodapé definidos pelo administrador deste *site* são utilizados em todos os outros *sites* de forma transparente e imediata.

Módulo `Quotes`

Com recurso à API do WordPress, este módulo fornece um tipo de conteúdo para a gestão de testemunhos dos estudantes da instituição. Estes testemunhos podem depois ser apresentados em áreas de conteúdo através do modelo definido no `ficheiro post-type-template-quotes.php`.

Modelos `post-type-template-featured.php`, `post-type-template-news.php` e `post-type-template-posts.php`

Definem vários modelos específicos para artigos utilizados em vários contextos no portal do IPLeiria. `posts` é o modelo geral utilizado para artigos, `featured` é o modelo utilizado para apresentação de artigos com imagem de destaque e `news` é o modelo utilizado para a apresentação de artigos com o respetivo excerto.

4.9.4. Tema `ipleiria_protocols`

Durante o ciclo de desenvolvimento do projeto foi solicitada a criação de um espaço para registo dos protocolos entre o IPLeiria e outras instituições. Neste sistema deveriam ser disponibilizadas as ferramentas para a gestão deste tipo de recursos, com suporte a vários campos incluindo campos do tipo ficheiro. No âmbito deste sistema foi criado o *micro-site* de

SGCPI

Protocolos do IPLeiria[47] ao qual foi associado este tema. As funcionalidades de gestão de protocolos são asseguradas pelo módulo `Protocols`, uma reimplementação do módulo `Courses` do tema ipleiria_courses adaptada às necessidades específicas deste projeto. Administrativamente foram também criados os campos personalizados necessários bem como definidos os modelos virtuais que estruturam o aspeto visual a apresentar aos visitantes. À dada de elaboração deste relatório estavam registados um pouco mais de 100 protocolos no sistema.

4.9.5. Tema `ipleiria_sdoc`

Os Serviços de Documentação do IPLeiria foram um dos primeiros serviços que mostrou interesse em migrar o *site* para o novo sistema, antes mesmo de a plataforma estar completamente desenvolvida. Para dar resposta a este novo objetivo, foi disponibilizado um novo *site* para o serviço[48] ao qual foi associado este tema. Para além da personalização visual associada às alterações de cores e logótipo do cabeçalho da página, foi implementado um novo módulo específico responsável pela gestão e apresentação de informação relacionada com a aquisição de recursos associadas às várias bibliotecas que geridas por este serviço. Este módulo `Acquisitions` assume-se, mais uma vez, como uma reimplementação do módulo `Courses` do tema ipleiria_courses.

4.10. Testes de desenvolvimento

4.10.1. Testes e análise de desempenho

No sentido de avaliar o desempenho da solução e identificar eventuais pontos de estrangulamento foram efetuados uma série de testes, incluindo algumas análises e medições

47 http://www.ipleiria.pt/protocolos/

48 http://www.ipleiria.pt/sdoc/

de baixo nível (*profiling*) ao tratamento e processamento de pedidos pela solução. Isto foi feito com recurso a uma instância virtual LAMP configurada com o Vagrant e o Puppet sob o VirtualBox, onde foi configurada a extensão Xdebug[49].

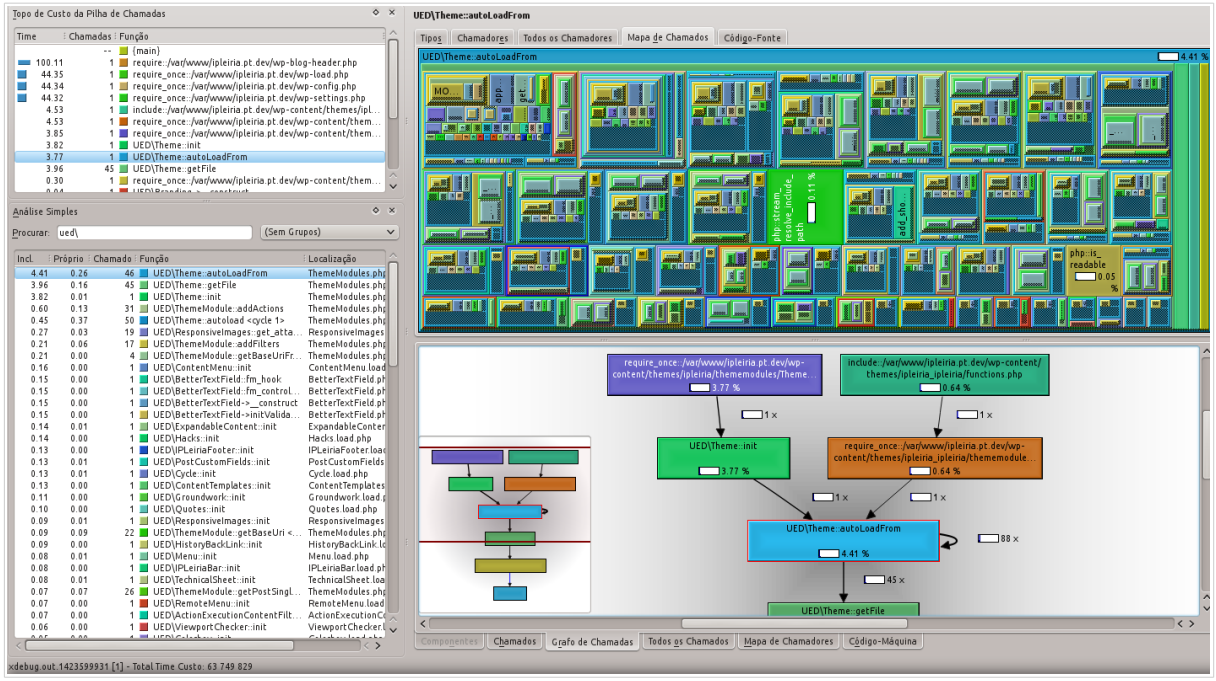


Figura 4.34: *Profiling* da solução

Desta análise foi identificado um ponto de estrangulamento no módulo RemoteMenu que, pelo número de chamadas à API do WordPress estava a condicionar o desempenho geral da solução. Após ter sido implementado um mecanismo de gestão da *cache* mais eficiente, o problema foi mitigado e assistiu-se a uma redução do tempo de resposta de cada pedido na ordem dos 50% (para um intervalo de 0,3 a 0,6 segundos – o tempo de resposta varia consoante a página e a complexidade estrutural da mesma).

49 http://xdebug.org/

4.10.2. Testes gráficos

No sentido de analisar o comportamento da solução em vários sistemas e dispositivos, foram realizados uma série de testes à componente visual da plataforma nos equipamentos e sistemas aos quais tivemos acesso:

- Sistemas operativos: Windows (XP, 7, 8 e 8.1), OSX, GNU/Linux (Ubuntu, Kubuntu, Mint, Debian, Arch Linux, etc.), Android, iOS;
- Navegadores Web: Mozilla Firefox, Google Chrome, Chromium, Internet Explorer (8, 9, 10), Safari, Links (navegador de texto);
- Forma: *Desktop*, portáteis, dispositivos móveis, *tablets* (ambiente virtualizado).

Estes testes permitiram detetar alguns problemas que foram sendo resolvidos ao longo do ciclo de desenvolvimento da solução.

No final dos ciclos de desenvolvimento foram também realizados testes de acessibilidade em ferramentas de validação automáticas como o AccessMonitor[50] da Unidade ACESSO da FCT ou o WAVE[51], tendo-se obtidos resultados finais muito positivos relativamente a este aspeto.

50 http://www.acessibilidade.gov.pt/accessmonitor/

51 http://wave.webaim.org

Página: http://www.ipleiria.pt/
Título: Politécnico de Leiria
Tamanho: 55.8 KB *(57164 bytes)*
Número de Elementos: 588
Data/Hora: 24/03/2015 - 12:01 GMT

Resultados compilados

I. Sumário

O índice que encontra no *AccessMonitor* é uma unidade de valoração utilizada em todos os testes do validador e cujo resultado final sintetiza e **quantifica o nível de acessibilidade alcançado**. O índice está representado numa escala de 1 a 10, representando o valor 10 uma adopção plena da boa prática induzida pelo *AccessMonitor*. **O índice é um indicador que se destina ao uso exclusivo dos criadores do sítio Web.** Todos os testes do *AccessMonitor* têm a sua fundamentação nas *WCAG 2.0* do *W3C*.

Esta página passa a bateria de testes do *AccessMonitor* de nivel "AAA"

índice AccessMonitor 10

Nivel	Testes realizados			
	Ok	Erros	Avisos	Total
A	1	0	8	9
AA	0	0	3	3
AAA	0	0	3	3

Figura 4.35: Resultados da bateria de testes do AccessMonitor

5. Colocação em produção

Após a implementação de toda a solução, tal como foi descrito na secção anterior, chega o momento de colocar o sistema em produção, isto é, disponibilizar a solução aos utilizadores. Nesta secção é descrita a arquitetura definida para a infraestrutura de produção, bem como a configuração do ambiente fornecido pela equipa de operações e parametrização da solução desenvolvida. Descrevem-se também as etapas seguintes onde já foram envolvidos utilizadores, nomeadamente a inserção e organização dos conteúdos iniciais no sistema, a apresentação do protótipo para um grupo de utilizadores internos, formação e apoio aos utilizadores e disponibilização da solução ao público em geral.

5.1. Arquitetura da infraestrutura de produção

Depois dos vários ciclos de testes e de desenvolvimento, que permitiram marcar a solução como estável para colocação em produção, foi necessário definir a arquitetura do sistema de suporte ao serviço. Este trabalho envolveu vários departamentos técnicos da área de informática do IPLeiria, numa equipa multi-disciplinar e polivalente que propôs a infraestrutura apresentada na Figura 5.1.

SGCPI

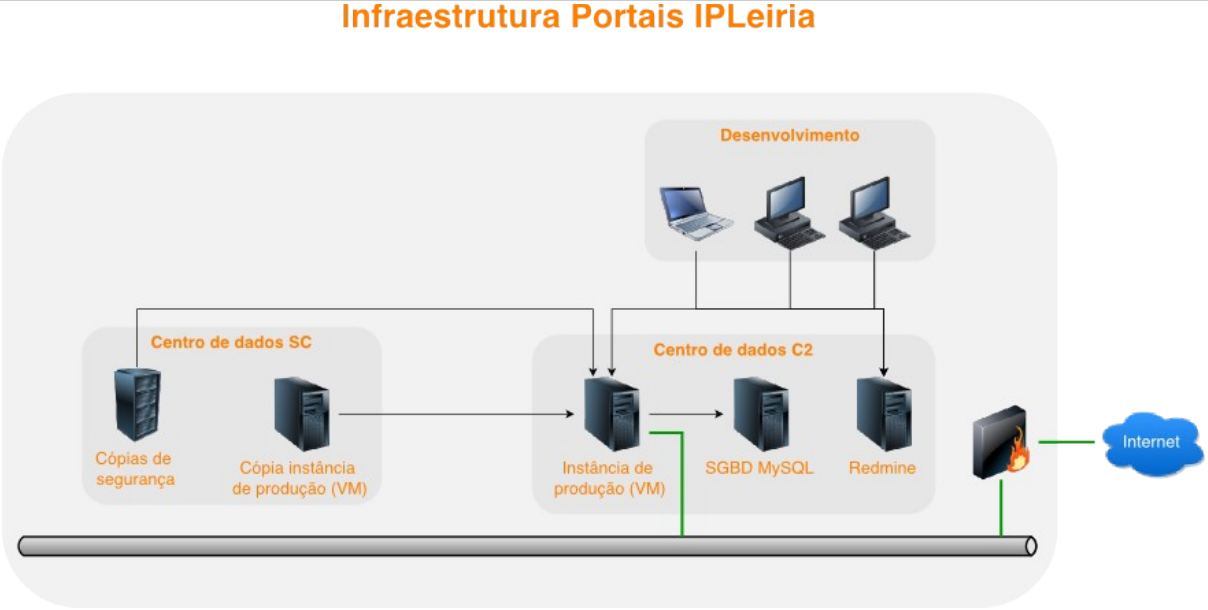


Figura 5.1: Infraestrutura tecnológica de suporte aos portais do IPLeiria

Nesta arquitetura temos uma instância de produção em ambiente virtual onde um sistema GNU/Linux (CentOS[52]) com um serviço HTTP fornecido pelo Apache e uma instância *multi-site* baseada em WordPress com a solução, fornece uma resposta a cada um dos pedidos HTTP efetuados ao serviço. A base de dados de suporte ao sistema é disponibilizada num *cluster* MySQL independente da solução. Diariamente é feita uma cópia binária dos *bits* alterados na instância de produção para uma cópia desta instância para um outro centro de dados, permitindo que em alguns minutos se possa iniciar uma instância com uma réplica do dia anterior, caso surja esta necessidade. Diariamente são também executadas cópias de segurança incrementais à solução, sendo semanalmente feitas cópias de segurança integrais, até ao limite de espaço alocado para as cópias de segurança no sistema de cópias de segurança. O código-fonte da solução está centralizado nos repositórios do projeto no serviço interno Redmine. Para além dos sistemas de proteção ativos na própria instância de produção, existe ainda uma *firewall* a proteger a rede interna e os serviços da instituição.

52 http://www.centos.org/

5.2. Configuração do ambiente e parametrização da solução

Depois do cenário estar implementado, a equipa de infraestrutura forneceu as credenciais de acesso ao sistema onde o sistema CentOS estava configurado, com o servidor HTTP Apache, *firewall* iptables, serviço SSH, contas de utilizador, bem como uma base dados para a solução no *cluster* MySQL para a qual também foram fornecidas as credenciais de acesso.

Numa primeira fase procedeu-se à instalação e parametrização do WordPress com os dados fornecidos, num endereço temporário (http://ww3.ipleiria.pt). O sistema foi configurado na pasta indicada pela equipa de infraestrutura (para facilitar o processo de cópias de segurança), com suporte para *multi-site* baseado em sub-diretórios, isto é cada um dos *sites* estaria acessível através de um endereço com o formato http://www.ipleiria.pt/[nomedosite], onde `[nomedosite]` seria substituído pelo nome de cada um dos *sites* a criar na solução. Esta abordagem tem várias vantagens:

- Continua a existir um isolamento entre os vários *sites* que compõem a solução, mas a transição entre *sites* passa a ser transparente para o utilizador, principalmente quando se mantém a consistência gráfica, tal como se previa nos requisitos do projeto;
- Torna muito mais simples e rápida a criação de novos *sites*, reduzindo a burocracia associada à gestão dos sub-domínios;
- Apesar do sistema adotar o formato de sub-diretórios, continua a ser possível criar *sites* baseados em sub-domínios se esta necessidade se vier a manifestar.

Apesar do endereço do provisório da solução ser público, foi ativada uma extensão no *site* que aplicava uma máscara que não permita o acesso ao *site* a utilizadores não autenticados. Aos utilizadores anónimos era apresentada uma página com uma mensagem a remeter para o *site*

antigo, sendo discretamente disponibilizada uma pequena ligação (no canto inferior direito desta página) que permitia o acesso ao formulário de autenticação, para que os utilizadores autorizados tivessem um acesso normal ao sistema. Depois dos utilizadores estarem autenticados, a visualização do *site* era feita sem qualquer tipo de restrições, tendo acesso completo à área pública, tal como seria apresentado ao visitante após o lançamento do sistema.

Do ponto de vista de segurança, foi feito o *hardening* da solução, tanto no WordPress como no próprio Apache, com o objetivo de reduzir o número de vetores de ataque. No caso do WordPress, foram retificadas as permissões, limitado o acesso a ficheiros sensíveis, alterado o prefixo das tabelas de dados, entre outros procedimentos. No Apache foi instalada e parametrizada a *firewall* aplicacional Web ModSecurity[53] (para filtragem de pedidos ao serviço), bem como o ModEvasive (para limitação da exposição a ataques de negação ao serviço).

Com o objetivo de aumentar o desempenho da solução foram ainda implementadas e/ou testadas algumas soluções para o Apache:

- Módulo mod_deflate[54] - permite ativar a compressão de dados na resposta do servidor, reduzindo o *overhead* associado às transferências de alguns recursos.

- Módulo mod_expires[55] - define alguns cabeçalhos HTTP utilizados para permitir a gestão da *cache* de alguns recursos estáticos nos navegadores clientes e em sistemas de *cache*. Foi utilizado para reduzir o volume de transferências de recursos entre pedidos consecutivos garantindo e maximizando a capacidade de resposta do servidor.

53 https://www.modsecurity.org/

54 http://httpd.apache.org/docs/2.2/mod/mod_deflate.html

55 http://httpd.apache.org/docs/2.2/mod/mod_expires.html

- Módulo pagespeed[56] - este módulo permite aplicar algumas técnicas de otimização de recursos e de *sites*, tais como redução e combinação de múltiplos ficheiros de um mesmo tipo, otimização de imagens, criação e gestão de *caches* de recursos, entre muitas outras funcionalidades. Após a implementação de algumas correções para compatibilizar a solução com este módulo, verificou-se que o código gerado pelo módulo apresentava problemas em questões relacionadas com a acessibilidade, onde era gerado código não conforme com as WCAG 2.0. A ativação deste módulo ficou adiada para uma segunda fase até que estas questões relacionadas com a acessibilidade estejam resolvidas.

Após a parametrização da solução foram feitos alguns testes de *stress* à solução para verificar a capacidade de resposta da mesma, tendo-se concluído que a mesma estava pronta para os primeiros testes de inserção de conteúdos.

5.3. Inserção dos conteúdos iniciais

Depois da solução estar configurada, foi dado acesso ao sistema a um grupo restrito de utilizadores para inserção dos conteúdos iniciais. Estes gestores de conteúdos foram responsáveis por importar e formatar os conteúdos tal como estava definido nas propostas gráficas fornecidas pelo *designer*. Esta equipa de utilizadores era composta por um grupo misto de pessoas com várias valências, desde programadores, passando por gestores de conteúdos, *designers*, diretores, administradores de sistemas, etc. Isto permitiu reunir um conjunto de opiniões e detetar alguns problemas menores que foram sendo prontamente resolvidos.

Paralelamente a este processo foi instalada e parametrizada uma solução para recolha de dados anónimos na área pública do sistema relacionados com as visitas dos utilizadores à

56 https://developers.google.com/speed/pagespeed/module

plataforma. Anteriormente esta análise era feita com recurso ao Google Analytics[57], mas esta solução tem limites associados às contas gratuitas, tem uma compatibilidade muito centrada nos próprios produtos e não permite um controlo absoluto sobre o tipo de informação que se pretende recolher, fornecendo uma API um pouco restritiva para a definição e acompanhamento de campanhas de *e-marketing*. No âmbito de uma análise às soluções existentes para *Web Analytics*, identificou-se o Piwik[58] como uma solução *open-source* de grande qualidade que permitiria reter o controlo dos dados recolhidos no âmbito deste projeto. Este sistema foi implementado e disponibilizado no endereço http://analytics.ipleiria.pt/, tendo sido devidamente integrado com a solução desenvolvida.

5.4. Apresentação do protótipo à comunidade

Após a inserção dos conteúdos iniciais, foi enviada uma mensagem à comunidade de funcionários docentes e não docentes por parte presidência do IPLeiria, a apresentar o trabalho desenvolvido ao longo dos vários meses do projeto e a solicitar à comunidade opiniões, sugestões e críticas, para ser disponibilizada uma solução com que todos se identificassem.

Foram registadas centenas de visitas e foram recebidas várias dezenas de comentários com sugestões e observações que foram triadas por níveis de prioridade e atribuídas às várias equipas envolvidas no projeto. Verificou-se que a maioria das observações estavam relacionadas com questões de conteúdos desatualizados ou de pormenores gráficos, situações que foram resolvidas num espaço de tempo relativamente curto.

57 http://www.google.com/analytics/

58 http://piwik.org/

5.5. Formação e apoio à gestão de conteúdos

Após a divulgação interna, foram identificadas pela presidência da instituição as equipas responsáveis pela gestão da informação de cada uma das secções dos novos portais, aos quais a equipa técnica associou os respetivos elementos com privilégios de edição. Nesta fase foram organizados *workshops* onde a equipa de desenvolvimento e de gestão de conteúdos se deslocou ao local de trabalho de cada equipa, onde se reuniu com cada um dos elementos para sensibilizar e formar as pessoas para o modo como deveria ser feita a gestão de conteúdos no novo sistema. O *feedback* obtido nestes trabalhos foi muito positivo e as pessoas mostraram-se muito motivadas com o trabalho desenvolvido.

Após estas formações iniciais, os utilizadores assumiram a gestão de conteúdos, tendo a equipa de desenvolvimento e de gestão de conteúdos assumindo o papel suporte técnico, acompanhando os trabalhos efetuados pelos utilizadores e esclarecendo dúvidas relacionadas com a gestão dos conteúdos. Ao fim de alguns dias os utilizadores tornaram-se praticamente autónomos, concluindo a atualização inicial dos conteúdos em cerca de duas semanas.

Na última semana antes do lançamento, articulou-se com o Gabinete de Imagem de Comunicação do IPLeiria a promoção de *teasers* baseados na imagem disponível no Anexo XXII. Isto servia para gerar algum interesse e expetativa nas redes sociais.

Figura 5.2: *Teaser* promocional no Facebook

5.6. Disponibilização ao público

Depois da formação e atualização dos conteúdos, a solução finalizada foi validada pela presidência do IPLeiria que autorizou a colocação da solução no endereço final e a divulgação pública do projeto.

Para o efeito o sistema foi colocado em modo de manutenção, foi implementada a reconfiguração do endereço e o sistema foi disponibilizado ao público em geral ao início da tarde do dia 9 de Fevereiro de 2015.

6. Resultados e conclusões

Este projeto apresentou uma solução tecnológica de suporte à construção de portais institucionais à medida, com a adaptação da solução ao caso particular do portais institucionais do IPLeiria. Pretendeu-se mostrar como uma plataforma *open-source*, amplamente utilizada no mercado pode ser adaptada às necessidades mais específicas de uma instituição, sem comprometer o funcionamento e as vantagens da própria solução, numa arquitetura modular e altamente extensível.

A informação registada pela plataforma de *Web Analytics* configurada permite-nos concluir que, nos primeiros 30 dias o sistema foi visitado 52270 vezes, com uma duração média de 7 minutos por cada visita. Foram vistas 226936 páginas, efetuadas 1428 pesquisas e transferidos 10554 ficheiros a partir do novo sistema. Se compararmos estes valores com os valores para o mesmo período do ano anterior sob o sistema antigo, temos um número de visitas inferior (72443 visitas, justificadas pelo facto de vários sistemas da rede do IPLeiria ainda estarem configurados com o endereço do sistema antigo), mas temos uma número de páginas visitadas superior (226936 *versus* 172268 do sistema anterior). Em média os utilizadores ficam mais 5 minutos em cada página do que no sistema SharePoint (2:07 *versus* 7:25 minutos) e novo sistema apresenta uma taxa de rejeição menor (visitantes que abandonaram o *site* após a primeira página – 57,4% do sistema antigo para 36% do novo sistema).

Isto significa que, numa análise grosseira dos números, os visitantes interagem mais com o novo sistema, durante um período de tempo superior e com uma taxa de abandono inferior. Isto corresponde aos comentários ao *feedback* que foi sendo recebido pelas redes sociais e sistemas de suporte onde os utilizadores se mostraram, na maioria dos casos, agradados com o

novo *site*, considerando que foi um salto qualitativo positivo relativamente à solução anterior. Por outro lado, verificou-se que os gestores de conteúdos adotaram o novo sistema sem grande resistência, assegurando a gestão da informação de forma autónoma e sem grande intervenção da equipa técnica, o que demonstra que a solução cumpre o objetivo de ser simples e funcional.

Verificou-se também que os *front-end developers* responsáveis pelo desenvolvimento de novos *sites* no sistema conseguiam fazê-lo sem grandes dificuldades, mesmo que fosse necessário desenvolver funcionalidades específicas para dar resposta a alguma necessidade particular de algum projeto. Comprovou-se que a solução proposta apresenta uma arquitetura suficientemente flexível para o desenvolvimento rápido de portais para a Web, onde o sistema de modelos virtuais e campos personalizados mostrou ser uma grande mais valia para a definição de estruturas específicas que acabam por ser transparentes para os gestores de conteúdo.

Nos dias seguintes à publicação da nova solução, a presidência identificou os próximos projetos estratégicos a implementar sob a solução, o que indica um claro interesse em continuar a apostar neste novo sistema. Estes novos sub-projetos dizem respeito à criação dos portais institucionais para as cinco escolas do IPLeiria, portal para o departamento de engenharia automóvel, portal de divulgação da oferta formativa exclusiva para estudantes chineses, bem como a criação de um diretório com informações sobre os funcionários docentes e não docentes da instituição.

Tecnicamente este projeto foi bastante desafiante, não só pela complexidade do problema como pela dimensão e pluralidade dos vários utilizadores que iriam assumir a gestão dos conteúdos, o que obrigava à implementação de uma solução que precisava de ser simples para utilizadores com um menor domínio da tecnologia, mas flexível o suficiente para permitir a implementação de funcionalidades mais avançadas para concretizar as tendências da Web do momento. Os resultados obtidos mostram que esta é, não só, uma solução perfeitamente

válida, como tem imenso potencial, permitindo que esta possa continuar a evoluir ao longo do tempo. Comprovou-se que se conseguiu desenvolver uma solução que melhorou os processos associados ao desenvolvimento e gestão da informação de portais Web.

Como trabalhos futuros, funcionalmente, temos a implementação de um assistente que permita a criação e implementação de estruturas e elementos compatíveis com o GroundworkCSS, implementação de um mecanismo para introdução de códigos adicionais no cabeçalho ou rodapé das páginas a partir da área administrativa, implementação de um mecanismo auditoria para registo de alterações de conteúdos de baixo nível (inserções, atualizações e eliminações de conteúdos por SQL), implementação de um mecanismo de carregamento de informação assíncrono para listas de conteúdos, implementação de uma extensão de autenticação baseada no ADFS, entre muitas outras funcionalidades. Estrategicamente o projeto vai continuar, sabendo-se já que nos próximos meses serão criados dezenas de novos *sites* sob a solução, no sentido de satisfazer as necessidades que estão a ser neste momento identificadas pela presidência do instituto. Tecnicamente esta é apenas a primeira versão de um projeto que continuará a crescer, onde futuras iterações da tecnologia irão continuar a melhorar a solução tornando-a num sistema cada vez mais capaz para a criação de portais institucionais.

Assim sendo, o saldo final parece ser manifestamente positivo, onde foi possível não só desenhar e implementar uma solução inovadora num espaço de tempo relativamente curto, como testar a mesma num cenário real onde passou a ser utilizada como o sistema de referência para a criação de portais Web. Pela dimensão e complexidade do problema, equipa envolvida e os resultados positivos obtidos, este projeto provou ser um trabalho extremamente motivante e moralizante, naquilo que posso considerar o culminar perfeito para o meu Mestrado em Engenharia Informática.

7. Bibliografia

[1]: Internet Live Stats, Number of Internet Users, Internet Live Stats. 2014. <http://www.internetlivestats.com/internet-users/>. [Acedido a 9 de Março de 2015]

[2]: ETC Digital, Digital Trends Demographics, ETC Digital. 2014. <http://etc-digital.org/digital-trends/consumer-behaviour/demographics/>. [Acedido a 10 de Março de 2015]

[3]: Internet Live Stats, Internet Users by Country, Internet Live Stats. 2014. <http://www.internetlivestats.com/internet-users-by-country/>. [Acedido a 9 de Março de 2015]

[4]: Marcotte, E., Responsive Web design, A List Apart. 2010. <http://www.alistapart.com/articles/responsive-web-design/>. [Acedido a 11 de Março de 2015]

[5]: Wikipedia contributors, Search engine optimization, Wikipedia, The Free Encyclopedia. 2015. <http://en.wikipedia.org/w/index.php?title=Search_engine_optimization&oldid=650068288>. [Acedido a 11 de Março de 2015]

[6]: 2015, February 2015 Web Server Survey, Netcraft Ltd. . <http://news.netcraft.com/archives/2015/02/24/february-2015-web-server-survey.html>. [Acedido a 11 de Março de 2015]

[7]: , Technologies Overview, W3Techs - Web Technology Surveys. 2015. <http://w3techs.com/technologies>. [Acedido a 11 de Março de 2015]

[8]: Rosso, Sara, WordPress Security White Paper, 2015

[9]: BuiltWith® Pty Ltd, CMS Usage Statistics, BuiltWith® Pty Ltd. . <http://trends.builtwith.com/cms>. [Acedido a 11 de Março de 2015]

[10]: Caldwell, B., Cooper, M. , Reid, L. & Vanderheiden, G., Web Content Accessibility Guidelines (WCAG) 2.0, W3C. 2008. <http://www.w3.org/TR/WCAG20/>. [Acedido a 12 de Março de 2015]

[11]: Craig, J. & Cooper, M., Accessible Rich Internet Applications (WAI-ARIA) 1.0, W3C. . <http://www.w3.org/TR/wai-aria/>. [Acedido a 12 de Março de 2015]

[12]: ISO/IEC/IEEE, Systems and software engineering - Vocabulary. ISO. 2010.

[13]: Jacob, Henry, Design Driven Development, D3 Foundation. 2007. <http://www.designdrivendevelopment.org/explore/index.html>. [Acedido a 11 de Março de 2015]

[14]: Drupal contributors, Multi-site - Sharing the same code base, Drupal. 2014. <https://www.drupal.org/documentation/install/multi-site#security>. [Acedido a 12 de Março de 2015]

[15]: Codex contributors, Plugin API, WordPress.org. 2015. <http://codex.wordpress.org/Plugin_API>. [Acedido a 12 de Março de 2015]

[16]: McFarlin, T., Using WordPress for Web Application Development: Rethinking Architecture, tuts+. 2013. <http://code.tutsplus.com/tutorials/using-wordpress-for-web-application-development-rethinking-architecture--wp-33880>. [Acedido a 12 de Março de 2015]

[17]: Eppstein, Christopher, Compass Documentation, compass-style.org. 2014. <http://compass-style.org/>. [Acedido a 19 de Março de 2015]

[18]: Harisov, Vitaly & Khachaturov, Leonid, CSSO, BEM. 2014. <https://en.bem.info/tools/optimizers/csso/>. [Acedido a 19 de Março de 2015]

[19]: Kovalyov, Anton, JSHint, a JavaScript Code Quality Tool, . 2015. <http://jshint.com/about/>. [Acedido a 19 de Março de 2015]

[20]: Hepting, Gary, GroundworkCSS 2 - A Responsive HTML5, CSS & Javascript Framework, GroundworkCSS 2. 2014. <https://groundworkcss.github.io/groundwork/docs/home.html>. [Acedido a 20 de Março de 2015]

[21]: Mozilla Developer Network & individual contributors, Window.postMessage(). Mozilla Developer Network. 2015.

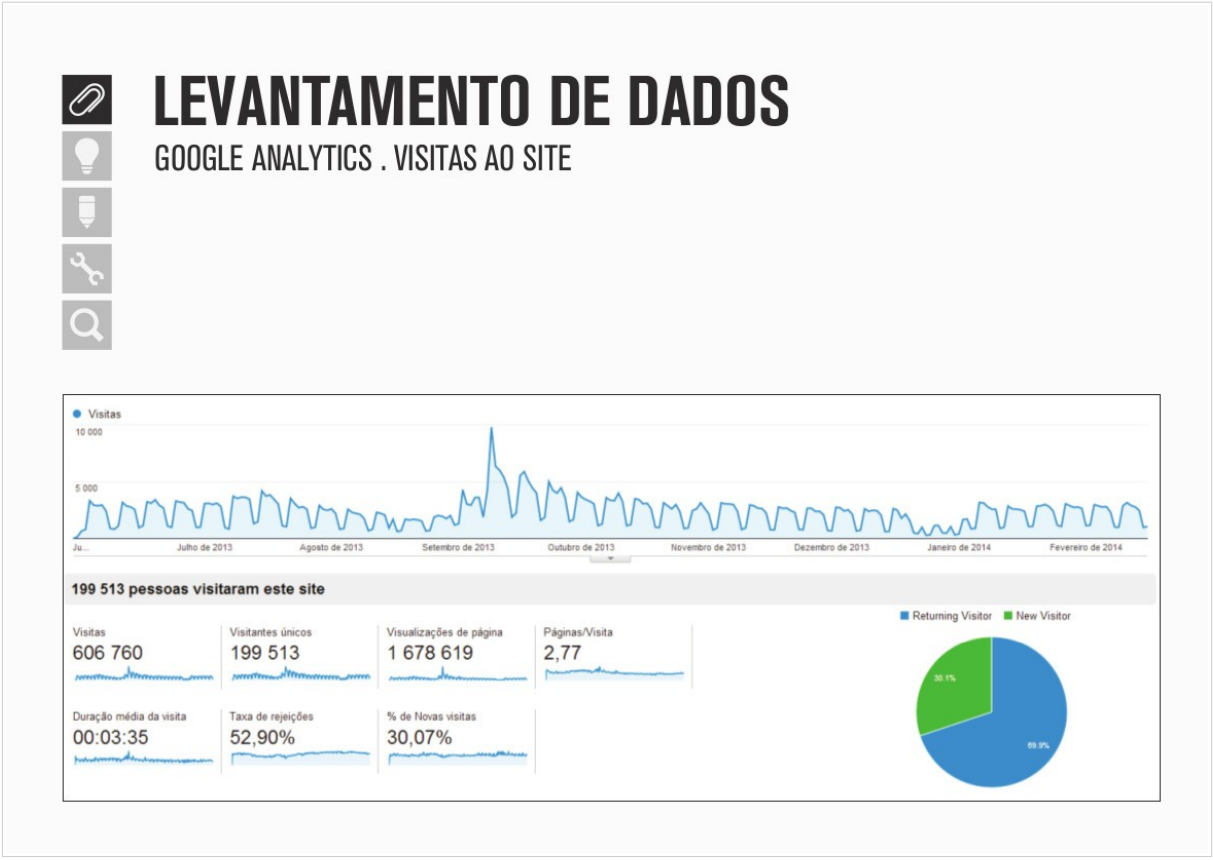
[22]: Berners-Lee, et al, URI Generic Syntax. The Internet Society. 2005.

8. Apêndices e anexos

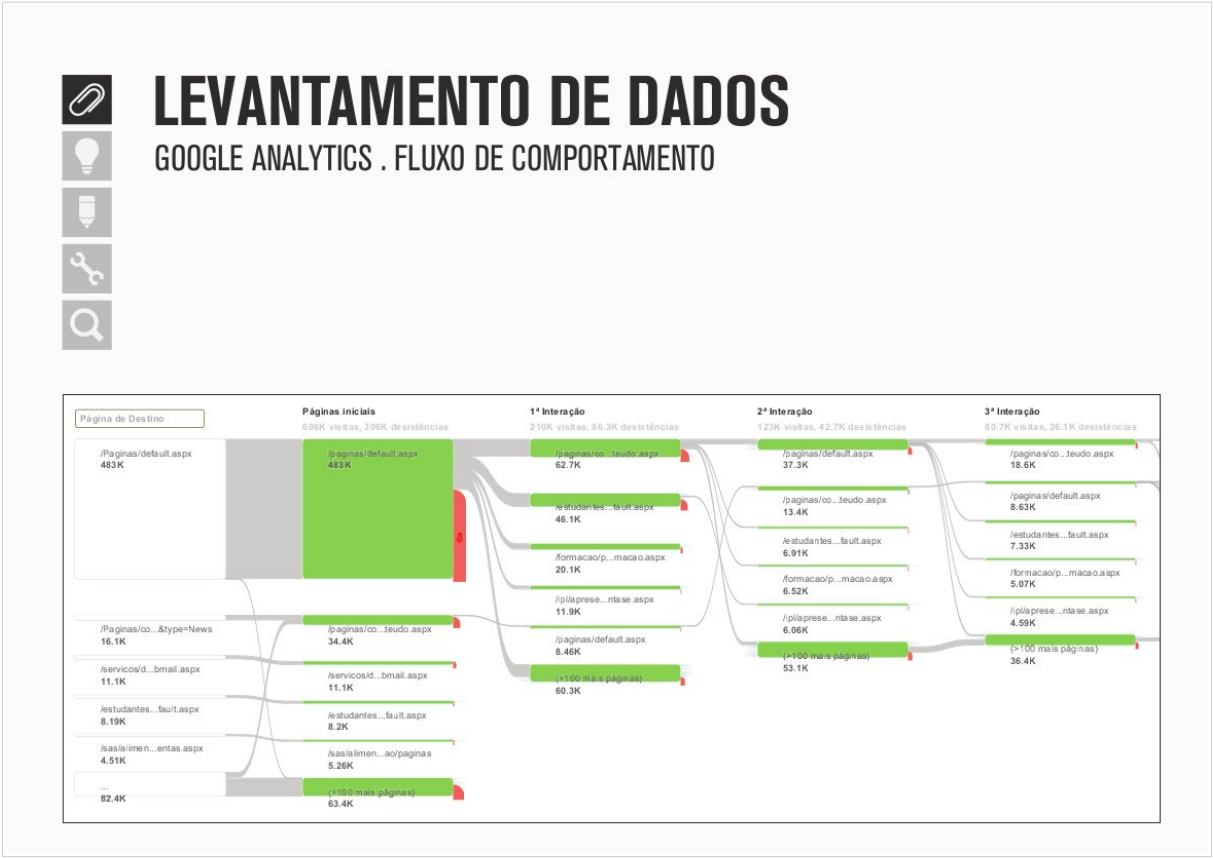
Anexo I - Necessidades, estratégias e objetivos da análise de dados ao *site* antigo


LEVANTAMENTO DE DADOS
IPLEIRIA
NECESSIDADES
Levantamento de toda a informação
Relevância da informação disponibilizada
Melhorar a navegação/usabilidade
ESTRATÉGIAS
Estratégias de comunicação
Novas tendências do mercado
Identificar necessidades de negócio
Determinar meios prioritários

Anexo II - Análise das visitas ao *site* antigo


LEVANTAMENTO DE DADOS
GOOGLE ANALYTICS . VISITAS AO SITE
Visitas
10 000
5 000
Julho de 2013
Agosto de 2013
Setembro de 2013
Outubro de 2013
Novembro de 2013
Dezembro de 2013
Janeiro de 2014
Fevereiro de 2014
199 513 pessoas visitaram este site
Visitas
606 760
Visitantes únicos
199 513
Visualizações de página
1 678 619
Páginas/Visita
2,77
Duração média da visita
00:03:35
Taxa de rejeições
52,90%
% de Novas visitas
30,07%
Returning Visitor
New Visitor

Anexo III - Comportamento dos visitantes no *site* antigo


LEVANTAMENTO DE DADOS
GOOGLE ANALYTICS . FLUXO DE COMPORTAMENTO
Página de Destino
Páginas iniciais
606K visitas, 396K desistências
1ª Interação
210K visitas, 86.3K desistências
2ª Interação
123K visitas, 42.7K desistências
3ª Interação
80.7K visitas, 26.1K desistências
/Paginas/default.aspx
483K
/Paginas/co...&type=News
16.1K
/servicos/d...bmail.aspx
11.1K
/estudantes...fault.aspx
8.19K
/sas/alimen...entas.aspx
4.51K
...
82.4K
/paginas/default.aspx
483K
/paginas/co...teudo.aspx
34.4K
/servicos/d...bmail.aspx
11.1K
/estudantes...fault.aspx
8.2K
/sas/alimen...ao/paginas
5.26K
(>100 mais páginas)
63.4K
/paginas/co...teudo.aspx
62.7K
/estudantes...fault.aspx
46.1K
/formacao/p...macao.aspx
20.1K
/pl/aprese...ntase.aspx
11.9K
/paginas/default.aspx
8.46K
(>100 mais páginas)
60.3K
/paginas/default.aspx
37.3K
/paginas/co...teudo.aspx
13.4K
/estudantes...fault.aspx
6.91K
/formacao/p...macao.aspx
6.52K
/pl/aprese...ntase.aspx
6.06K
(>100 mais páginas)
53.1K
/paginas/co...teudo.aspx
18.6K
/paginas/default.aspx
8.63K
/estudantes...fault.aspx
7.33K
/formacao/p...macao.aspx
5.07K
/pl/aprese...ntase.aspx
4.59K
(>100 mais páginas)
36.4K

Anexo IV - Idioma/países dos visitantes no *site* antigo

LEVANTAMENTO DE DADOS

GOOGLE ANALYTICS . IDIOMA PAÍS/TERRITÓRIO

	Idioma	Visitas	% Visitas
1.	pt-pt	358 152	59,03%
2.	pt	152 597	25,15%
3.	pt-br	41 342	6,81%
4.	en-us	33 271	5,48%
5.	en	6 536	1,08%
6.	es	2 398	0,40%
7.	en-gb	1 956	0,32%
8.	fr	1 233	0,20%
9.	ru	1 018	0,17%
10.	zh-cn	994	0,16%

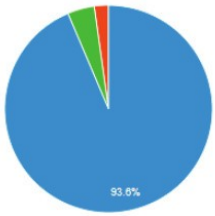
	País/Território	Visitas	% Visitas
1.	Portugal	583 307	96,30%
2.	Brazil	4 862	0,80%
3.	Spain	2 263	0,37%
4.	France	1 161	0,19%
5.	United States	1 156	0,19%
6.	United Kingdom	1 090	0,18%
7.	Italy	849	0,14%
8.	Germany	815	0,13%
9.	Poland	723	0,12%
10.	Switzerland	710	0,12%

Anexo V - Dispositivos de acesso ao *site* antigo

LEVANTAMENTO DE DADOS

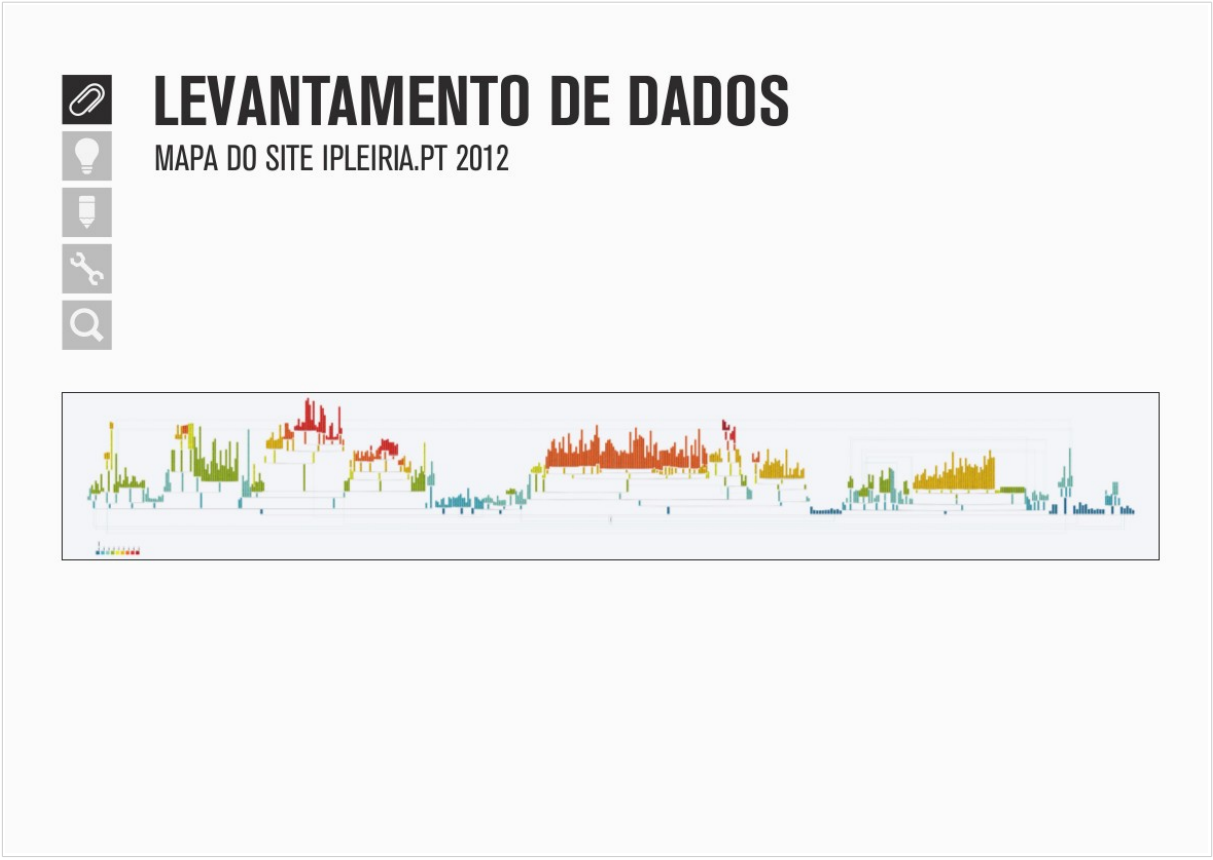
GOOGLE ANALYTICS . DISPOSITIVOS

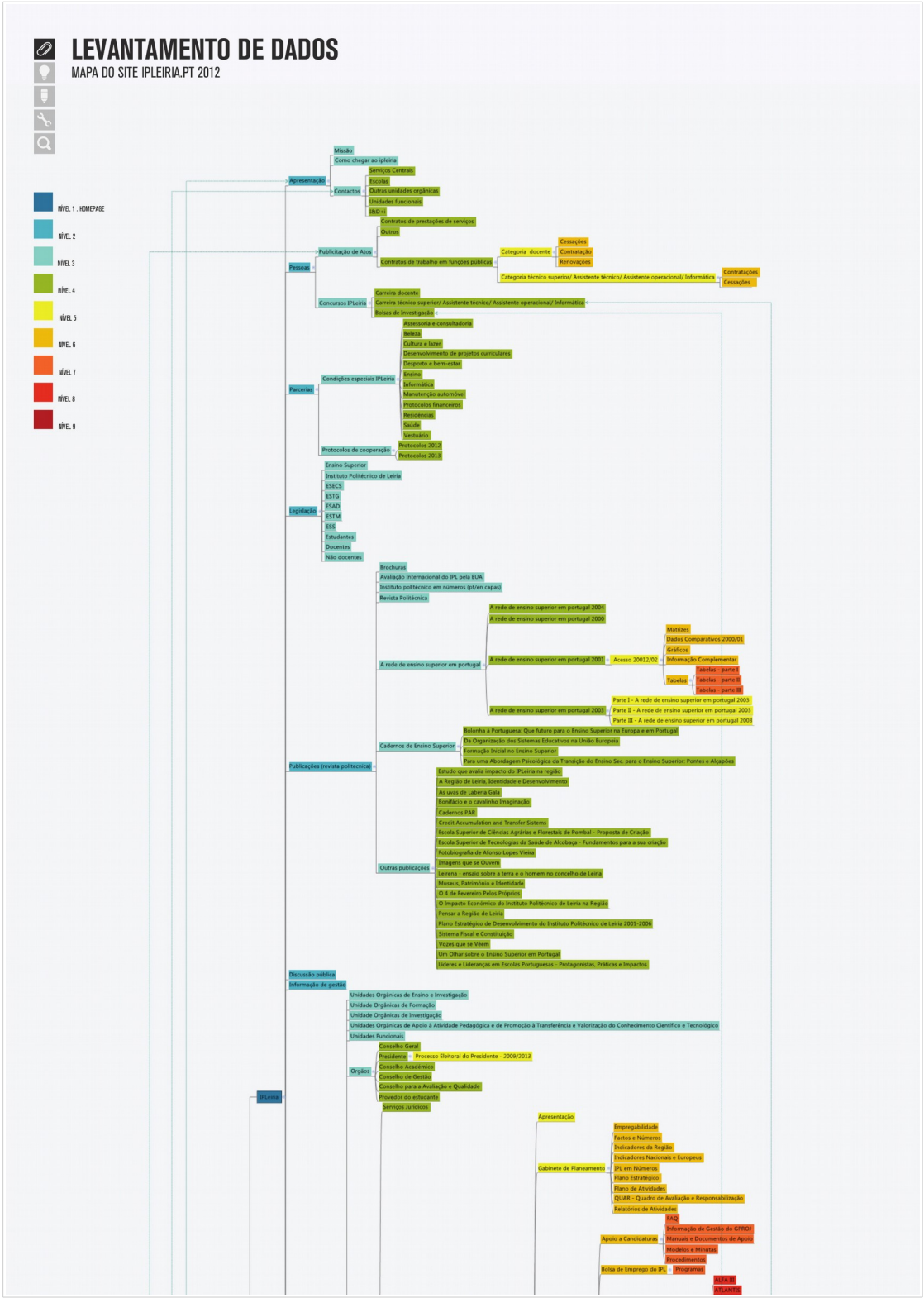
Categoria do Dispositivo	Visitas	Visitas	Contribuição para o total: Visitas
	605 701 % do total: 99,83% (606 760)	605 701 % do total: 99,83% (606 760)	
1. desktop	567 018	93,61%	
2. mobile	25 324	4,18%	
3. tablet	13 359	2,21%	



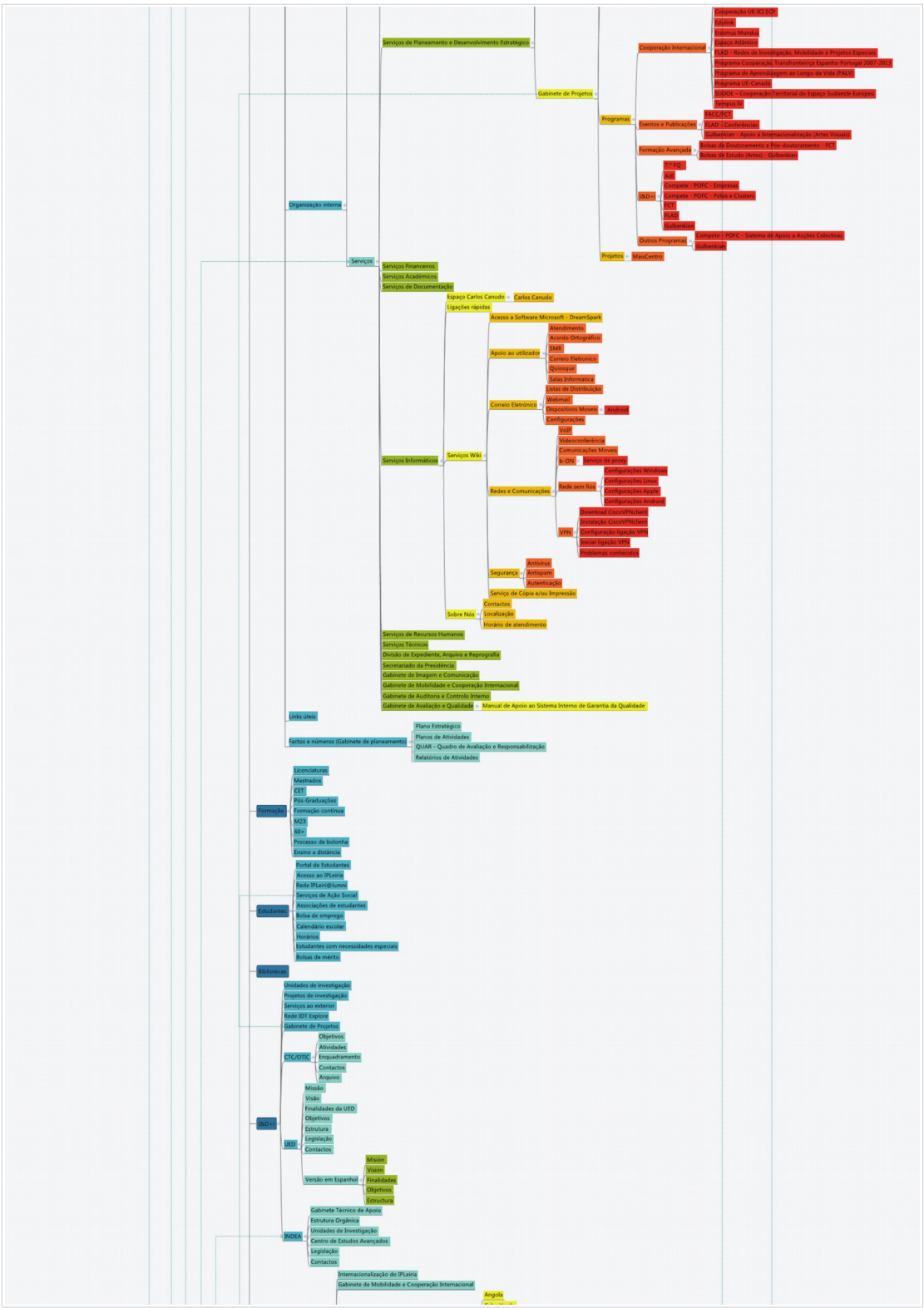
Anexo VI - Estrutura do menu do *site* anterior do IPLeiria

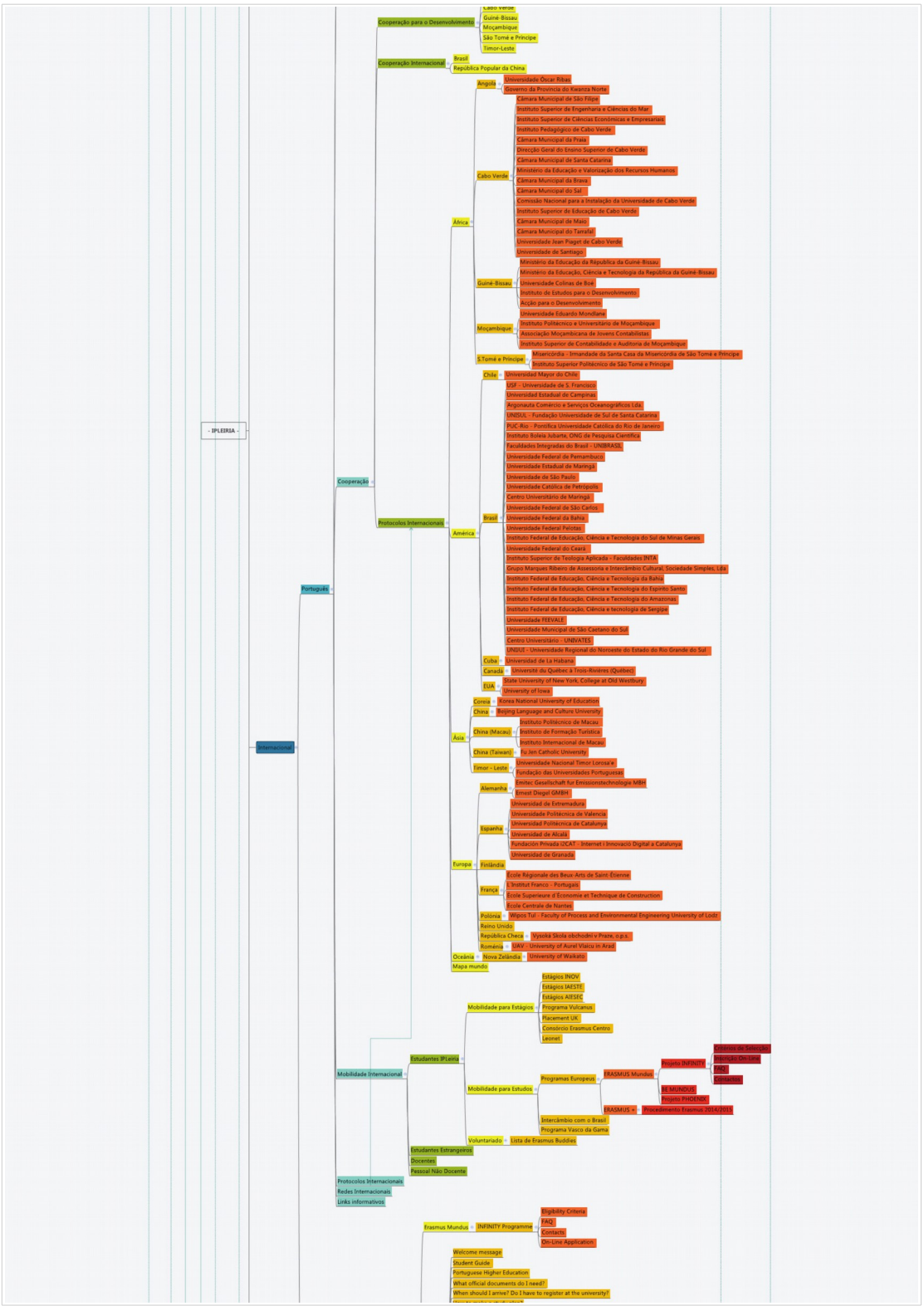
SGCPI



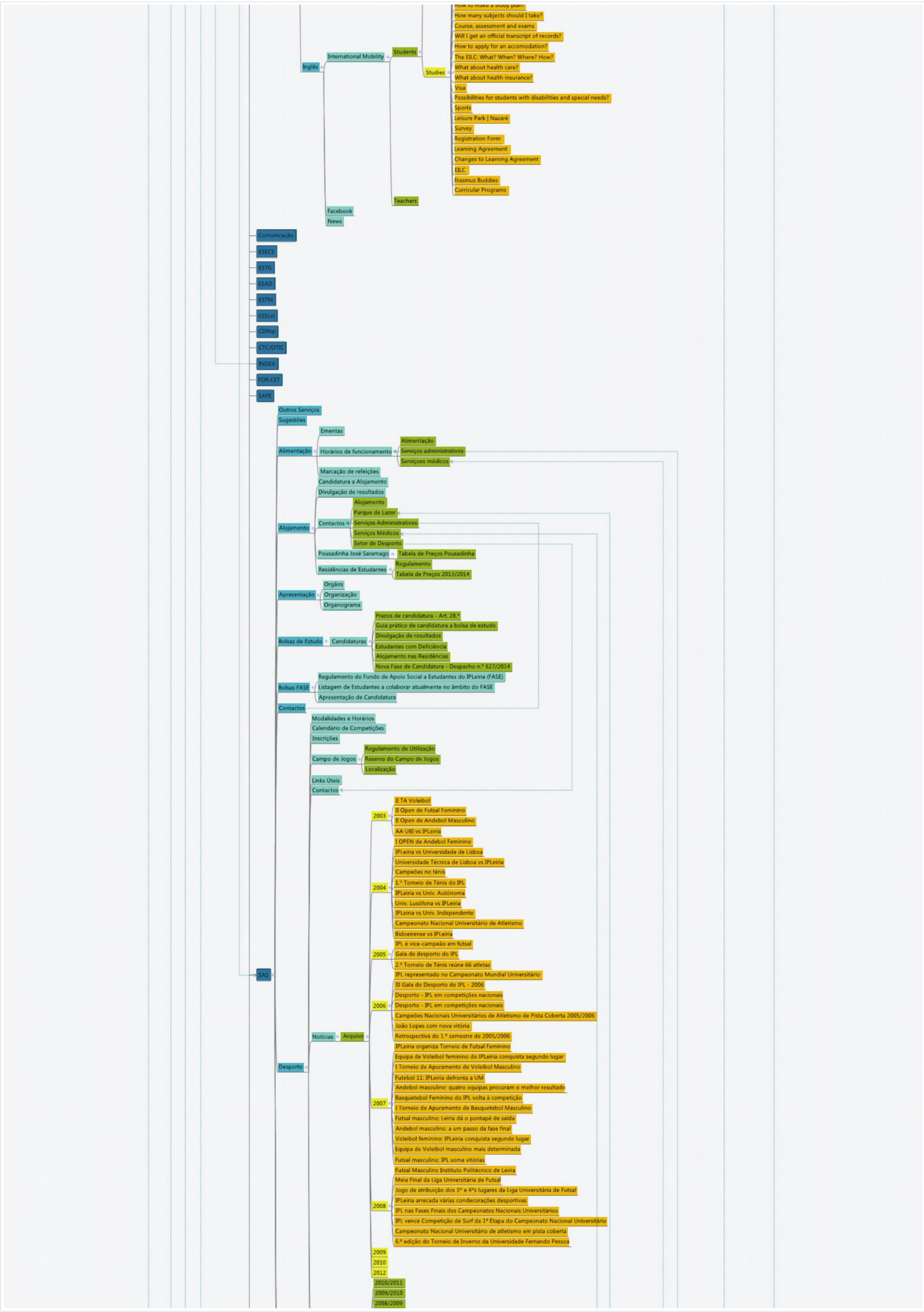

LEVANTAMENTO DE DADOS
MAPA DO SITE IPLEIRIA.PT 2012
NÍVEL 1 . HOMEPAGE
NÍVEL 2
NÍVEL 3
NÍVEL 4
NÍVEL 5
NÍVEL 6
NÍVEL 7
NÍVEL 8
NÍVEL 9
Missão
Como chegar ao ipleiria
Apresentação
Serviços Centrais
Escolas
Contactos
Outras unidades orgânicas
Unidades funcionais
I&D+I
Contratos de prestações de serviços
Outros
Publicitação de Atos
Pessoas
Contratos de trabalho em funções públicas
Categoria docente
Cessações
Contratação
Renovações
Categoria técnico superior/ Assistente técnico/ Assistente operacional/ Informática
Contratações
Cessações
Concursos IPLeiria
Carreira docente
Carreira técnico superior/ Assistente técnico/ Assistente operacional/ Informática
Bolsas de Investigação
Parcerias
Condições especiais IPLeiria
Assessoria e consultadoria
Beleza
Cultura e lazer
Desenvolvimento de projetos curriculares
Desporto e bem-estar
Ensino
Informática
Manutenção automóvel
Protocolos financeiros
Residências
Saúde
Vestuário
Protocolos de cooperação
Protocolos 2012
Protocolos 2013
Legislação
Ensino Superior
Instituto Politécnico de Leiria
ESECS
ESTG
ESAD
ESTM
ESS
Estudantes
Docentes
Não docentes
Publicações (revista politecnica)
Brochuras
Avaliação Internacional do IPL pela EUA
Instituto politécnico em números (pt/en capas)
Revista Politécnica
A rede de ensino superior em portugal
A rede de ensino superior em portugal 2004
A rede de ensino superior em portugal 2000
A rede de ensino superior em portugal 2001
Acesso 2001/2/02
Matrizes
Dados Comparativos 2000/01
Gráficos
Informação Complementar
Tabelas
Tabelas - parte I
Tabelas - parte II
Tabelas - parte III
A rede de ensino superior em portugal 2003
Parte I - A rede de ensino superior em portugal 2003
Parte II - A rede de ensino superior em portugal 2003
Parte III - A rede de ensino superior em portugal 2003
Cadernos de Ensino Superior
Bolonha à Portuguesa: Que futuro para o Ensino Superior na Europa e em Portugal
Da Organização dos Sistemas Educativos na União Europeia
Formação Inicial no Ensino Superior
Para uma Abordagem Psicológica da Transição do Ensino Sec. para o Ensino Superior: Pontes e Alçapões
Outras publicações
Estudo que avalia impacto do IPLeiria na região
A Região de Leiria, Identidade e Desenvolvimento
As uvas de Labéria Gala
Bonifácio e o cavalinho Imaginação
Cadernos PARI
Credit Accumulation and Transfer Sistems
Escola Superior de Ciências Agrárias e Florestais de Pombal - Proposta de Criação
Escola Superior de Tecnologias da Saúde de Alcobaça - Fundamentos para a sua criação
Fotobiografia de Afonso Lopes Vieira
Imagens que se Ouvem
Leirena - ensaio sobre a terra e o homem no concelho de Leiria
Museus, Património e Identidade
O 4 de Fevereiro Pelos Próprios
O Impacto Económico do Instituto Politécnico de Leiria na Região
Pensar a Região de Leiria
Plano Estratégico de Desenvolvimento do Instituto Politécnico de Leiria 2001-2006
Sistema Fiscal e Constituição
Vozes que se Vêem
Um Olhar sobre o Ensino Superior em Portugal
Líderes e Lideranças em Escolas Portuguesas - Protagonistas, Práticas e Impactos
Discussão pública
Informação de gestão
IPLeiria
Unidades Orgânicas de Ensino e Investigação
Unidade Orgânicas de Formação
Unidade Orgânicas de Investigação
Unidades Orgânicas de Apoio à Atividade Pedagógica e de Promoção à Transferência e Valorização do Conhecimento Científico e Tecnológico
Unidades Funcionais
Orgãos
Conselho Geral
Presidente
Processo Eleitoral do Presidente - 2009/2013
Conselho Académico
Conselho de Gestão
Conselho para a Avaliação e Qualidade
Provedor do estudante
Serviços Jurídicos
Gabinete de Planeamento
Apresentação
Empregabilidade
Factos e Números
Indicadores da Região
Indicadores Nacionais e Europeus
IPL em Números
Plano Estratégico
Plano de Atividades
QUAR - Quadro de Avaliação e Responsabilização
Relatórios de Atividades
Apoio a Candidaturas
FAQ
Informação de Gestão do GPEQI
Manuais e Documentos de Apoio
Modelos e Minutas
Procedimentos
Bolsa de Emprego do IPL
Programas
ALFA II
ATLANTIS

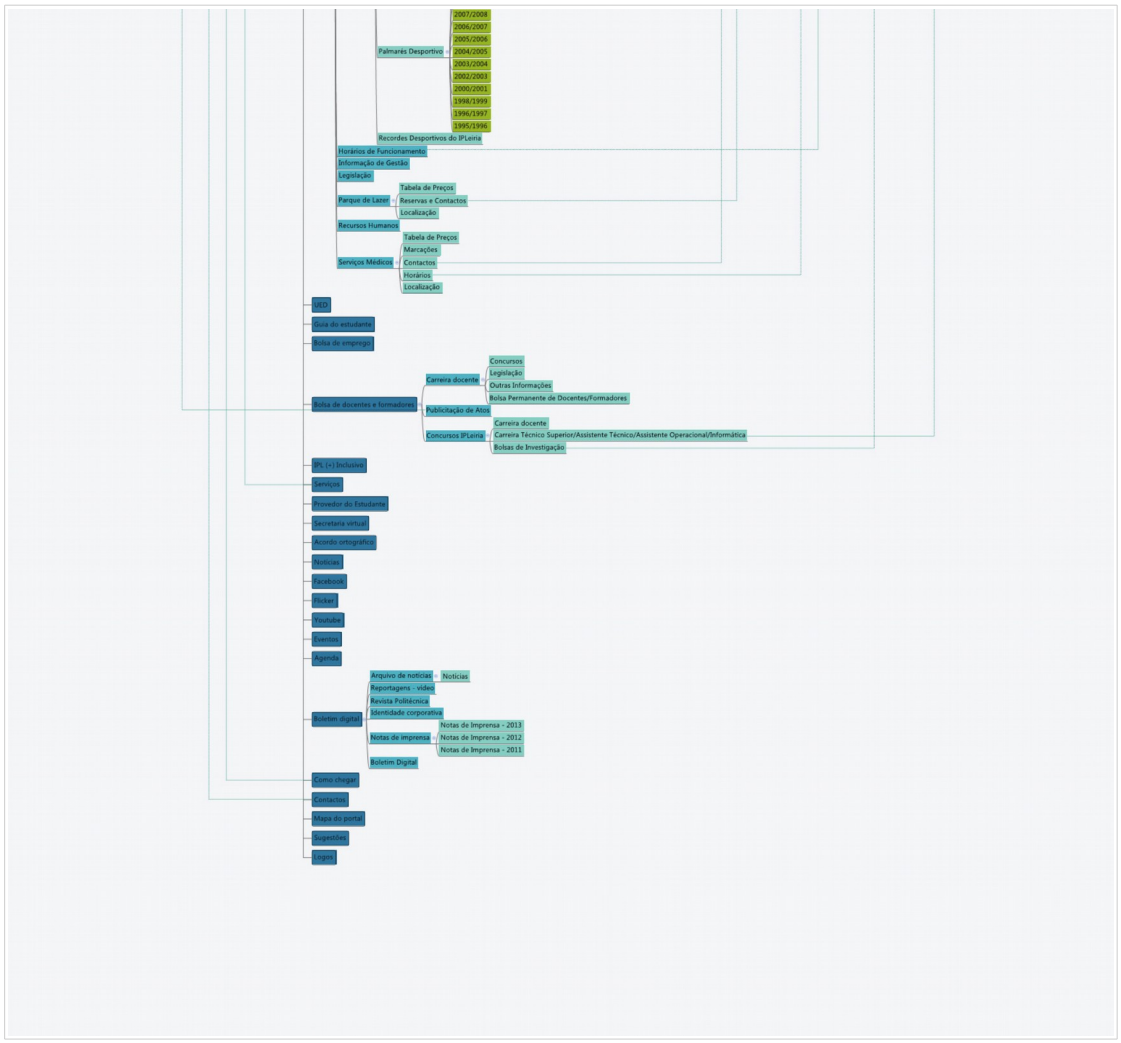
SGCPI


Serviços de Planeamento e Desenvolvimento Estratégico
Gabinete de Projetos
Cooperação Internacional
Cooperação UE-ICI ECP
EuroMed
Erasmus Mundus
Espaço Atlântico
FLAD – Redes de Investigação, Mobilidade e Projetos Especiais
Programa Cooperação Transfronteiriça Espanha-Portugal POCTEP 2007-2013
Programa de Aprendizagem ao Longo da Vida (PALV)
Programa UE-Canadá
SUDOE – Cooperação Territorial do Espaço Sudoeste Europeu
Tempus IV
Programas
Eventos e Publicações
FACC/FCT
FLAD : Conferências
Gulbenkian - Apoio à Internacionalização (Artes Visuais)
Formação Avançada
Bolsas de Doutoramento e Pós-doutoramento - FCT
Bolsas de Estudo (Artes) – Gulbenkian
I&D e I
7º PQ
Aidi
Compete - POFC - Empresas
Compete - POFC - Pólos e Clusters
FCT
FLAD
Gulbenkian
Outros Programas
Compete I POFC - Sistema de Apoio a Acções Colectivas
Gulbenkian
Projetos
MaisCentro
Organização interna
Serviços
Serviços Financeiros
Serviços Académicos
Serviços de Documentação
Serviços Informáticos
Espaço Carlos Canudo
Carlos Canudo
Ligações rápidas
Acesso a Software Microsoft - DreamSpark
Apoio ao utilizador
Atendimento
Acordo Ortográfico
SMR
Correio Eletronico
Quiosque
Salas Informatica
Correio Eletrónico
Listas de Distribuição
Webmail
Dispositivos Moveis
Android
Configurações
Serviços Wiki
Redes e Comunicações
VoIP
Videoconferência
Comunicações Moveis
b-ON
Serviço de proxy
Rede sem fios
Configurações Windows
Configurações Linux
Configurações Apple
Configurações Android
VPN
Download CiscoVPNClient
Instalação CiscoVPNClient
Configuração ligação VPN
Iniciar ligação VPN
Problemas conhecidos
Segurança
Antivírus
Antispam
Autenticação
Serviço de Cópia e/ou Impressão
Sobre Nós
Contactos
Localização
Horário de atendimento
Serviços de Recursos Humanos
Serviços Técnicos
Divisão de Expediente, Arquivo e Reprografia
Secretariado da Presidência
Gabinete de Imagem e Comunicação
Gabinete de Mobilidade e Cooperação Internacional
Gabinete de Auditoria e Controlo Interno
Gabinete de Avaliação e Qualidade
Manual de Apoio ao Sistema Interno de Garantia da Qualidade
Links úteis
Factos e números (Gabinete de planeamento)
Plano Estratégico
Planos de Atividades
QUAR - Quadro de Avaliação e Responsabilização
Relatórios de Atividades
Formação
Licenciaturas
Mestrados
CET
Pós-Graduações
Formação continua
M23
RVCC
Processo de bolonha
Ensino a distância
Estudantes
Portal de Estudantes
Acesso ao IPLeiria
Rede IPLeiri@fumos
Serviços de Ação Social
Associações de estudantes
Bolsa de emprego
Calendário escolar
Horários
Estudantes com necessidades especiais
Bolsas de mérito
Bibliotecas
Unidades de investigação
Projetos de investigação
Serviços ao exterior
Rede IDI Explore
Gabinete de Projetos
I&D / I
CTC/OTIC
Objetivos
Atividades
Enquadramento
Contactos
Arquivo
UID
Missão
Visão
Finalidades da UID
Objetivos
Estrutura
Legislação
Contactos
Versão em Espanhol
Misión
Visión
Finalidades
Objetivos
Estructura
INDEA
Gabinete Técnico de Apoio
Estrutura Orgânica
Unidades de Investigação
Centro de Estudos Avançados
Legislação
Contactos
Internacionalização do IPLeiria
Gabinete de Mobilidade e Cooperação Internacional
Angola


- IPLEIRIA -
Internacional
Português
Cooperação
Mobilidade Internacional
Protocolos Internacionais
Redes Internacionais
Links informativos
Cooperação para o Desenvolvimento
Cabo Verde
Guiné-Bissau
Moçambique
São Tomé e Príncipe
Timor-Leste
Cooperação Internacional
Brasil
República Popular da China
Protocolos Internacionais
África
Angola
Universidade Óscar Ribas
Governo da Província do Kwanza Norte
Cabo Verde
Câmara Municipal de São Filipe
Instituto Superior de Engenharia e Ciências do Mar
Instituto Superior de Ciências Económicas e Empresariais
Instituto Pedagógico de Cabo Verde
Câmara Municipal da Praia
Direcção Geral do Ensino Superior de Cabo Verde
Câmara Municipal de Santa Catarina
Ministério da Educação e Valorização dos Recursos Humanos
Câmara Municipal da Brava
Câmara Municipal do Sal
Comissão Nacional para a Instalação da Universidade de Cabo Verde
Instituto Superior de Educação de Cabo Verde
Câmara Municipal de Maio
Câmara Municipal do Tarrafal
Universidade Jean Piaget de Cabo Verde
Universidade de Santiago
Guiné-Bissau
Ministério da Educação da República da Guiné-Bissau
Ministério da Educação, Ciência e Tecnologia da República da Guiné-Bissau
Universidade Colinas de Boé
Instituto de Estudos para o Desenvolvimento
Acção para o Desenvolvimento
Universidade Eduardo Mondlane
Moçambique
Instituto Politécnico e Universitário de Moçambique
Associação Moçambicana de Jovens Contabilistas
Instituto Superior de Contabilidade e Auditoria de Moçambique
S.Tomé e Príncipe
Misericórdia - Irmandade da Santa Casa da Misericórdia de São Tomé e Príncipe
Instituto Superior Politécnico de São Tomé e Príncipe
América
Chile
Universidad Mayor do Chile
Brasil
USF - Universidade de S. Francisco
Universidad Estadual de Campinas
Argonauta Comércio e Serviços Oceanográficos Lda
UNISUL - Fundação Universidade de Sul de Santa Catarina
PUC-Rio - Pontifica Universidade Católica do Rio de Janeiro
Instituto Roleia Jubarte, ONG de Pesquisa Científica
Faculdades Integradas do Brasil - UNIBRASIL
Universidade Federal de Pernambuco
Universidade Estadual de Maringá
Universidade de São Paulo
Universidade Católica de Petrópolis
Centro Universitário de Maringá
Universidade Federal de São Carlos
Universidade Federal da Bahia
Universidade Federal Pelotas
Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais
Universidade Federal do Ceará
Instituto Superior de Teologia Aplicada - Faculdades INTA
Grupo Marques Ribeiro de Assessoria e Intercâmbio Cultural, Sociedade Simples, Lda
Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas
Instituto Federal de Educação, Ciência e tecnologia de Sergipe
Universidade FEEVALE
Universidade Municipal de São Caetano do Sul
Centro Universitário - UNIVATES
UNIJUÍ - Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Cuba
Universidad de La Habana
Canadá
Université du Québec à Trois-Rivières (Québec)
EUA
State University of New York, College at Old Westbury
University of Iowa
Ásia
Coreia
Korea National University of Education
China
Beijing Language and Culture University
China (Macau)
Instituto Politécnico de Macau
Instituto de Formação Turística
Instituto Internacional de Macau
China (Taiwan)
Fu Jen Catholic University
Timor - Leste
Universidade Nacional Timor Lorosa'e
Fundação das Universidades Portuguesas
Europa
Alemanha
Fintec Gesellschaft für Emissionstechnologie MBH
Ernest Diegel GMBH
Espanha
Universidad de Extremadura
Universidade Politécnica de Valencia
Universidad Politécnica de Catalunya
Universidad de Alcalá
Fundación Privada I2CAT - Internet i Innovació Digital a Catalunya
Universidad de Granada
Finlândia
França
École Régionale des Beux-Arts de Saint-Étienne
L'Institut Franco - Portugais
École Supérieure d'Économie et Technique de Construction
École Centrale de Nantes
Polónia
Wipos Tuf - Faculty of Process and Environmental Engineering University of Lodz
Reino Unido
República Checa
Vysoka Skola obchodni v Praze, o.p.s.
Roménia
UAV - University of Aurel Vlaicu in Arad
Oceânia
Nova Zelândia
University of Waikato
Mapa mundo
Mobilidade para Estágios
Estágios INOV
Estágios IAESTE
Estágios AIESEC
Programa Vulcanus
Placement UK
Consórcio Erasmus Centro
Leonel
Estudantes IPLeiria
Mobilidade para Estudos
Programas Europeus
ERASMUS Mundus
Projeto INFINITY
Critérios de Selecção
Inscrição On-Line
FAQ
Contactos
BE MUNDUS
Projeto PHOENIX
ERASMUS +
Procedimento Erasmus 2014/2015
Intercâmbio com o Brasil
Programa Vasco da Gama
Voluntariado
Lista de Erasmus Buddies
Estudantes Estrangeiros
Docentes
Pessoal Não Docente
Protocolos Internacionais
Redes Internacionais
Links informativos
Erasmus Mundus
INFINITY Programme
Eligibility Criteria
FAQ
Contacts
On-Line Application
Welcome message
Student Guide
Portuguese Higher Education
What official documents do I need?
When should I arrive? Do I have to register at the university?

SGCPI


How many subjects should I take?
Course, assessment and exams
Will I get an official transcript of records?
How to apply for an accomodation?
The EILC: What? When? Where? How?
What about health care?
What about health insurance?
Visa
Possibilities for students with disabilities and special needs?
Sports
Leisure Park | Nazaré
Survey
Registration Form
Learning Agreement
Changes to Learning Agreement
EILC
Erasmus Buddies
Curricular Programs
Studies
Students
Teachers
International Mobility
Inglês
Facebook
News
Comunicação
ESECS
ESTG
ESAD
ESTM
ESSLei
CDRsp
CTC/OTIC
INDEA
FOR.CET
SAPE
Outros Serviços
Sugestões
Ementas
Alimentação
Horários de funcionamento
Alimentação
Serviços administrativos
Serviçoes médicos
Marcação de refeições
Candidatura a Alojamento
Divulgação de resultados
Alojamento
Parque de Lazer
Serviços Administrativos
Serviços Médicos
Setor de Desporto
Contactos
Alojamento
Pousadinha José Saramago
Tabela de Preços Pousadinha
Residências de Estudantes
Regulamento
Tabela de Preços 2013/2014
Apresentação
Orgãos
Organização
Organograma
Bolsas de Estudo
Candidaturas
Prazos de candidatura - Art. 28.º
Guia prático de candidatura a bolsa de estudo
Divulgação de resultados
Estudantes com Deficiência
Alojamento nas Residências
Nova Fase de Candidatura - Despacho n.º 627/2014
Bolsas FASE
Regulamento do Fundo de Apoio Social a Estudantes do IPLeiria (FASE)
Listagem de Estudantes a colaborar atualmente no âmbito do FASE
Apresentação de Candidatura
Contactos
Modalidades e Horários
Calendário de Competições
Inscrições
Campo de Jogos
Regulamento de Utilização
Reserva do Campo de Jogos
Localização
Links Úteis
Contactos
SAS
Desporto
Notícias
Arquivo
2003
II TA Voleibol
II Open de Futsal Feminino
II Open de Andebol Masculino
AA UBI vs IPLeiria
2004
I OPEN de Andebol Feminino
IPLeiria vs Universidade de Lisboa
Universidade Técnica de Lisboa vs IPLeiria
Campeões no ténis
1.º Torneio de Ténis do IPL
IPLeiria vs Univ. Autónoma
Univ. Lusófona vs IPLeiria
IPLeiria vs Univ. Independente
Campeonato Nacional Universitário de Atletismo
Bidoeirense vs IPLeiria
2005
IPL é vice-campeão em futsal
Gala de desporto do IPL
2.º Torneio de Ténis reúne 66 atletas
2006
IPL representado no Campeonato Mundial Universitário
III Gala do Desporto do IPL - 2006
Desporto - IPL em competições nacionais
Desporto - IPL em competições nacionais
Campeões Nacionais Universitários de Atletismo de Pista Coberta 2005/2006
João Lopes com nova vitória
Retrospectiva do 1.º semestre de 2005/2006
2007
IPLeiria organiza Torneio de Futsal Feminino
Equipa de Voleibol feminino do IPLeiria conquista segundo lugar
I Torneio de Apuramento de Voleibol Masculino
Futebol 11: IPLeiria defronta a UM
Andebol masculino: quatro equipas procuram o melhor resultado
Basquetebol Feminino do IPL volta à competição
I Torneio de Apuramento de Basquetebol Masculino
Futsal masculino: Leiria dá o pontapé de saída
Andebol masculino: a um passo da fase final
Voleibol feminino: IPLeiria conquista segundo lugar
Equipa de Voleibol masculino mais determinada
Futsal masculino: IPL soma vitórias
2008
Futsal Masculino Instituto Politécnico de Leiria
Meia Final da Liga Universitária de Futsal
Jogo de atribuição dos 3º e 4ºs lugares da Liga Universitária de Futsal
IPLeiria arrecada várias condecorações desportivas
IPL nas Fases Finais dos Campeonatos Nacionais Universitários
IPL vence Competição de Surf da 1ª Etapa do Campeonato Nacional Universitário
Campeonato Nacional Universitário de atletismo em pista coberta
6.ª edição do Torneio de Inverno da Universidade Fernando Pessoa
2009
2010
2012
2010/2011
2009/2010
2008/2009


2007/2008
2006/2007
2005/2006
Palmarés Desportivo
2004/2005
2003/2004
2002/2003
2000/2001
1998/1999
1996/1997
1995/1996
Recordes Desportivos do IPLeiria
Horários de Funcionamento
Informação de Gestão
Legislação
Tabela de Preços
Parque de Lazer
Reservas e Contactos
Localização
Recursos Humanos
Tabela de Preços
Marcações
Serviços Médicos
Contactos
Horários
Localização
UED
Guia do estudante
Bolsa de emprego
Concursos
Carreira docente
Legislação
Outras Informações
Bolsa de docentes e formadores
Bolsa Permanente de Docentes/Formadores
Publicitação de Atos
Carreira docente
Concursos IPLeiria
Carreira Técnico Superior/Assistente Técnico/Assistente Operacional/Informática
Bolsas de Investigação
IPL (+) Inclusivo
Serviços
Provedor do Estudante
Secretaria virtual
Acordo ortográfico
Notícias
Facebook
Flickr
Youtube
Eventos
Agenda
Arquivo de notícias
Notícias
Reportagens - vídeo
Revista Politécnica
Identidade corporativa
Boletim digital
Notas de imprensa
Notas de Imprensa - 2013
Notas de Imprensa - 2012
Notas de Imprensa - 2011
Boletim Digital
Como chegar
Contactos
Mapa do portal
Sugestões
Login

Anexo VII - Identificação do perfil dos visitantes


LEVANTAMENTO DE DADOS
PERFIS DE UTILIZADORES
PÚBLICO EM GERAL
Imprensa
Inst. Ensino
Empresas
Investigadores
ANTIGOS ALUNOS (Alumni)
CET
Licenciaturas
Mestrados
Formação
ESTUDANTES
EaD
Presencial
Erasmus
CANDIDATOS
Nacionais
Internacionais
STAFF
Dirigentes
Docentes
Não Docentes

Anexo VIII - Proposta de menus para o novo *site*

SGCPI


IDEIAS / CONCEITOS
ENTRADAS NO MENU
MENU ACESSO RÁPIDO
REDE IPLEIRIA
LOG IN
MENU PERFIS DE UTILIZADOR
CANDIDATOS
ESTUDANTES
ALUMNI
DOCENTES
FUNCIONÁRIOS
EMPRESAS
MENU PRINCIPAL
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA

Anexo IX - Considerações da proposta e serviços a implementar


IDEIAS / CONCEITOS
SITE IPLEIRIA.PT 2014
TER EM CONTA
O público-alvo
Comunicação para os diferentes públicos
Regra dos 3 cliques (usabilidade)
Compatível com equipamentos móveis
SERVIÇOS
Criação e inserção de conteúdos
Redes Sociais
Responsive design
Acessibilidade
Multi-idioma

Anexo X - Considerações gráficas e funcionais do projeto

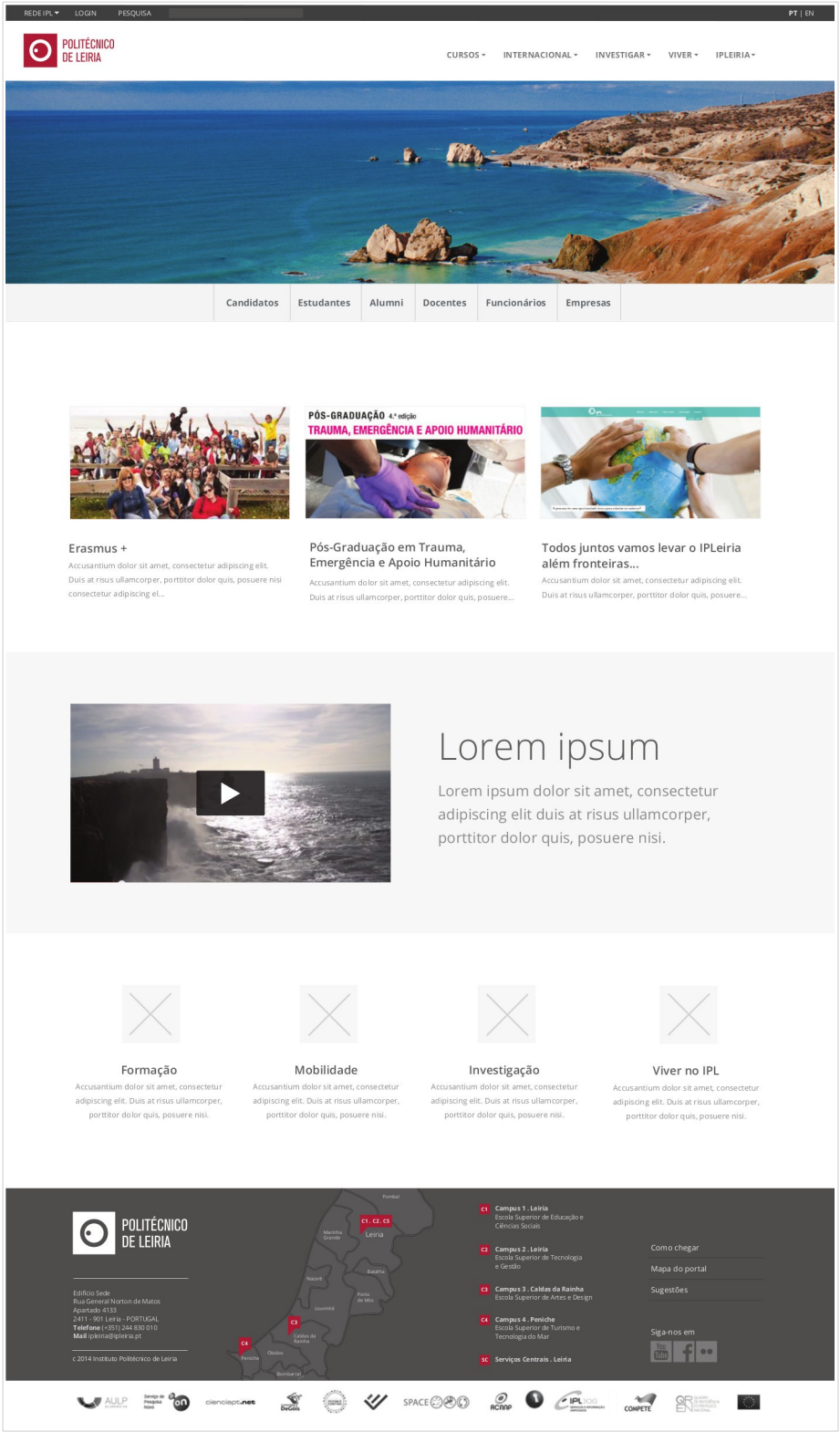

PROTÓTIPO
INTERFACE ACESSÍVEL E FÁCIL DE USAR
INTERFACE
Otimizar e facilitar a experiência do utilizador
Acesso à informação de forma rápida e eficaz
Facilitar ao utilizador a compreensão do site

Anexo XI - *Wireframes* e *mockups*


PROTÓTIPO
WIREFRAMES & MOCKUPS
NAVEGAÇÃO
Menu acesso rápido | Login | Pesquisa
Menu principal (geral ipleiria)
Menu secundário
Menu perfis de utilizador

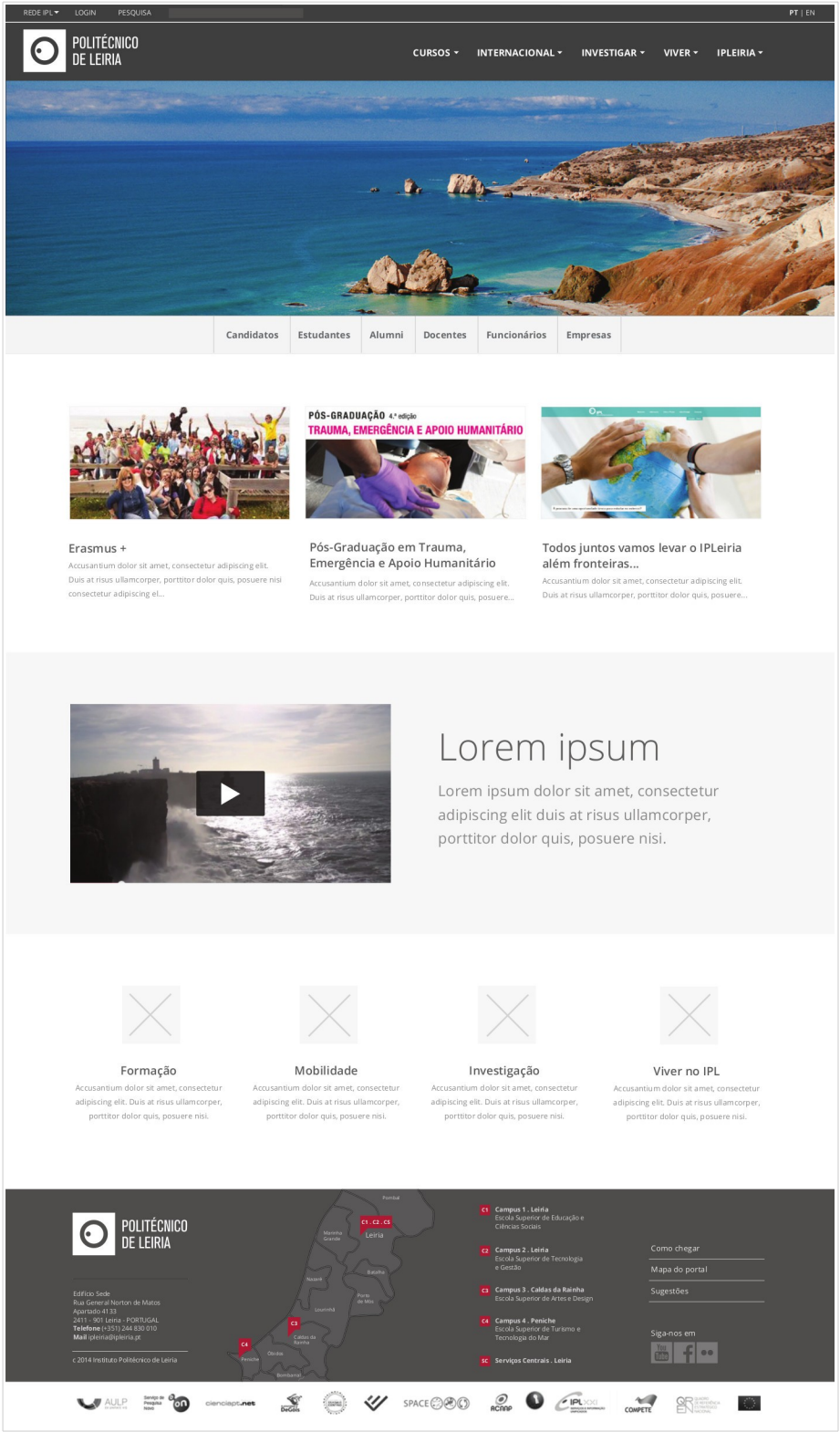
Anexo XII - Proposta gráfica 1

SGCPI


REDE IPL
LOGIN
PESQUISA
PT | EN
POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
Candidatos
Estudantes
Alumni
Docentes
Funcionários
Empresas
PÓS-GRADUAÇÃO 4.ª edição
TRAUMA, EMERGÊNCIA E APOIO HUMANITÁRIO
Erasmus +
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi consectetur adipiscing el...
Pós-Graduação em Trauma, Emergência e Apoio Humanitário
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Todos juntos vamos levar o IPLeiria além fronteiras...
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Formação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Mobilidade
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Investigação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Viver no IPL
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
POLITÉCNICO DE LEIRIA
Edifício Sede
Rua General Norton de Matos
Apartado 4133
2411 - 901 Leiria - PORTUGAL
Telefone (+351) 244 830 010
Mail ipleiria@ipleiria.pt
c 2014 Instituto Politécnico de Leiria
C1 Campus 1 . Leiria
Escola Superior de Educação e Ciências Sociais
C2 Campus 2 . Leiria
Escola Superior de Tecnologia e Gestão
C3 Campus 3 . Caldas da Rainha
Escola Superior de Artes e Design
C4 Campus 4 . Peniche
Escola Superior de Turismo e Tecnologia do Mar
SC Serviços Centrais . Leiria
Como chegar
Mapa do portal
Sugestões
Siga-nos em

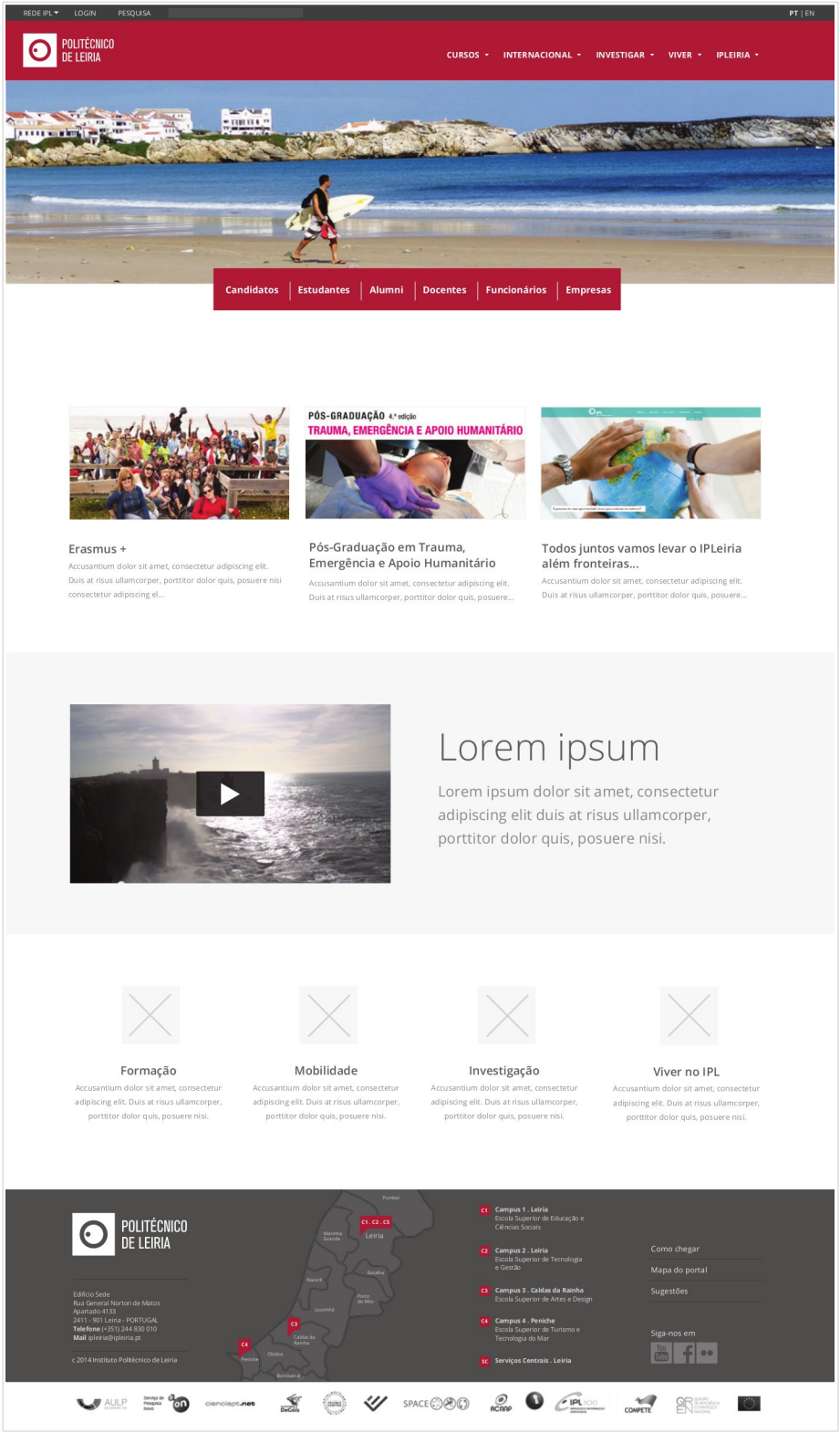
Anexo XIII - Proposta gráfica 2

SGCPI


REDE IPL
LOGIN
PESQUISA
PT | EN
POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
Candidatos
Estudantes
Alumni
Docentes
Funcionários
Empresas
PÓS-GRADUAÇÃO 4.ª edição
TRAUMA, EMERGÊNCIA E APOIO HUMANITÁRIO
Erasmus +
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi consectetur adipiscing el...
Pós-Graduação em Trauma, Emergência e Apoio Humanitário
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Todos juntos vamos levar o IPLeiria além fronteiras...
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Formação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Mobilidade
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Investigação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Viver no IPL
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
POLITÉCNICO DE LEIRIA
Edifício Sede
Rua General Norton de Matos
Apartado 4133
2411 - 901 Leiria - PORTUGAL
Telefone (+351) 244 830 010
Mail ipleiria@ipleiria.pt
c 2014 Instituto Politécnico de Leiria
C1 . C2 . C5
Leiria
C1 Campus 1 . Leiria
Escola Superior de Educação e Ciências Sociais
C2 Campus 2 . Leiria
Escola Superior de Tecnologia e Gestão
C3 Campus 3 . Caldas da Rainha
Escola Superior de Artes e Design
C4 Campus 4 . Peniche
Escola Superior de Turismo e Tecnologia do Mar
SC Serviços Centrais . Leiria
Como chegar
Mapa do portal
Sugestões
Siga-nos em

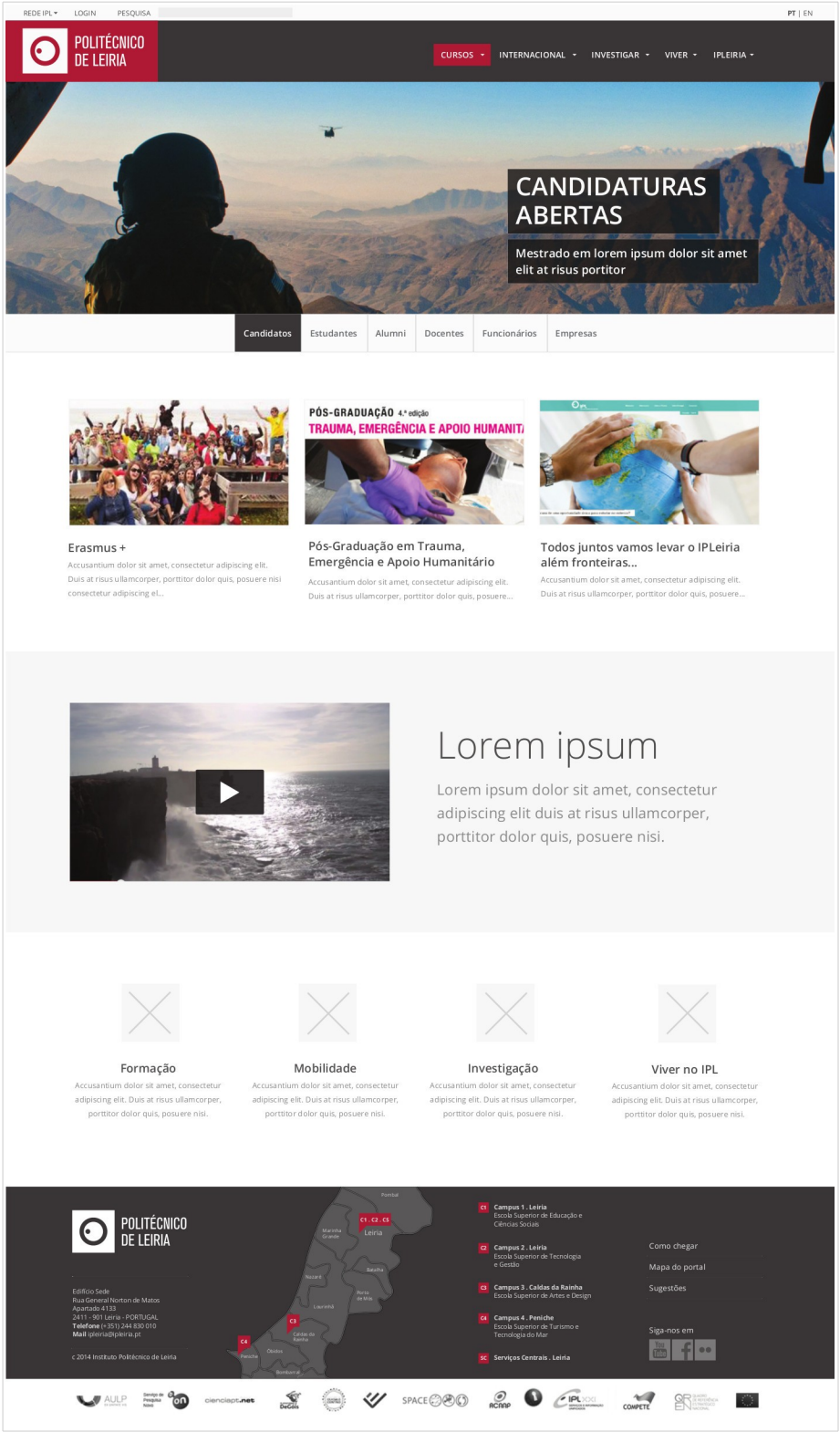
Anexo XIV - Proposta gráfica 3

SGCPI


REDE IPL
LOGIN
PESQUISA
PT | EN
POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
Candidatos
Estudantes
Alumni
Docentes
Funcionários
Empresas
PÓS-GRADUAÇÃO 4.ª edição
TRAUMA, EMERGÊNCIA E APOIO HUMANITÁRIO
Erasmus +
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi consectetur adipiscing el...
Pós-Graduação em Trauma, Emergência e Apoio Humanitário
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Todos juntos vamos levar o IPLeiria além fronteiras...
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Formação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Mobilidade
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Investigação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Viver no IPL
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
POLITÉCNICO DE LEIRIA
Edifício Sede
Rua General Norton de Matos
Apartado 4133
2411 - 901 Leiria - PORTUGAL
Telefone (+351) 244 830 010
Mail ipleria@ipleiria.pt
© 2014 Instituto Politécnico de Leiria
C1 Campus 1 . Leiria
Escola Superior de Educação e Ciências Sociais
C2 Campus 2 . Leiria
Escola Superior de Tecnologia e Gestão
C3 Campus 3 . Caldas da Rainha
Escola Superior de Artes e Design
C4 Campus 4 . Peniche
Escola Superior de Turismo e Tecnologia do Mar
SC Serviços Centrais . Leiria
Como chegar
Mapa do portal
Sugestões
Siga-nos em

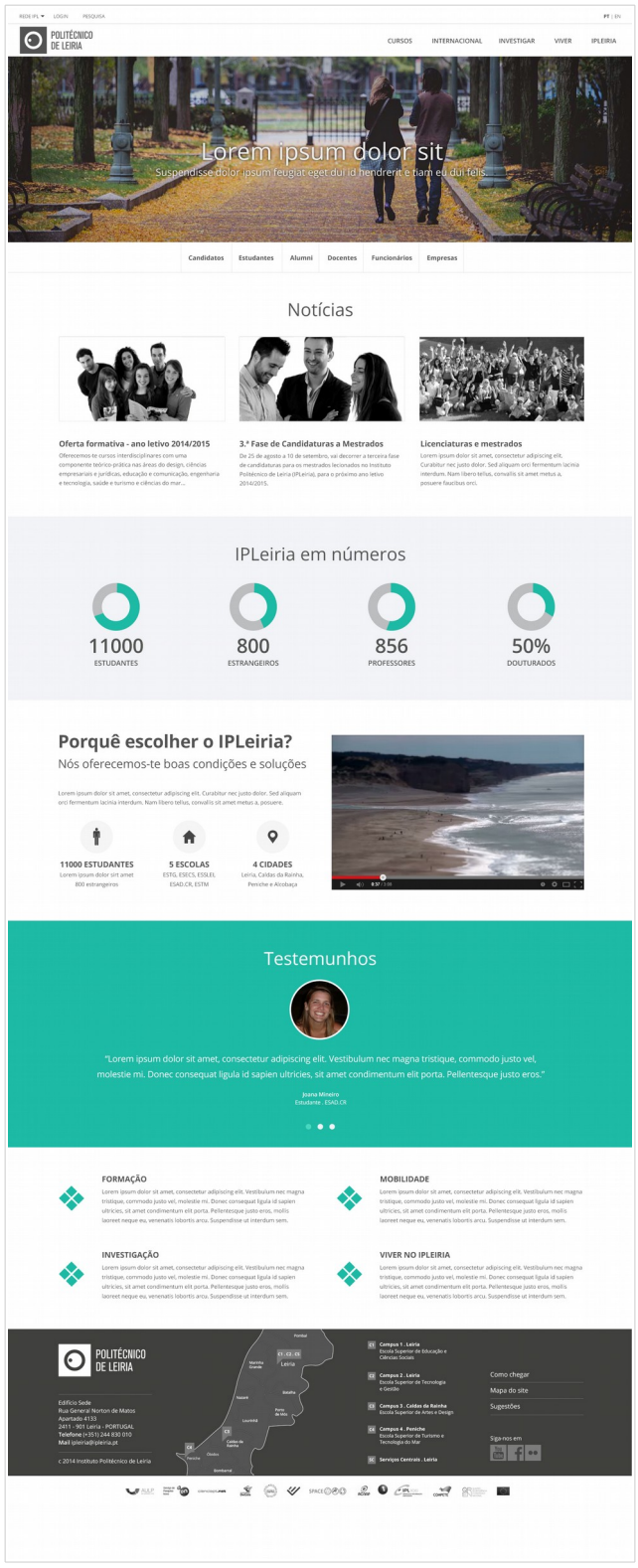
Anexo XV - Proposta gráfica 4

SGCPI


REDE IPL
LOGIN
PESQUISA
PT | EN
POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
CANDIDATURAS ABERTAS
Mestrado em lorem ipsum dolor sit amet elit at risus portitor
Candidatos
Estudantes
Alumni
Docentes
Funcionários
Empresas
Erasmus +
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi consectetur adipiscing el...
PÓS-GRADUAÇÃO 4.ª edição
TRAUMA, EMERGÊNCIA E APOIO HUMANITÁ
Pós-Graduação em Trauma, Emergência e Apoio Humanitário
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Todos juntos vamos levar o IPLeiria além fronteiras...
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere...
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Formação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Mobilidade
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Investigação
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
Viver no IPL
Accusantium dolor sit amet, consectetur adipiscing elit. Duis at risus ullamcorper, porttitor dolor quis, posuere nisi.
POLITÉCNICO DE LEIRIA
Edifício Sede
Rua General Norton de Matos
Apartado 4133
2411 - 901 Leiria - PORTUGAL
Telefone (+351) 244 830 010
Mail ipleiria@ipleiria.pt
c 2014 Instituto Politécnico de Leiria
Campus 1 . Leiria
Escola Superior de Educação e Ciências Sociais
Campus 2 . Leiria
Escola Superior de Tecnologia e Gestão
Campus 3 . Caldas da Rainha
Escola Superior de Artes e Design
Campus 4 . Peniche
Escola Superior de Turismo e Tecnologia do Mar
Serviços Centrais . Leiria
Como chegar
Mapa do portal
Sugestões
Siga-nos em

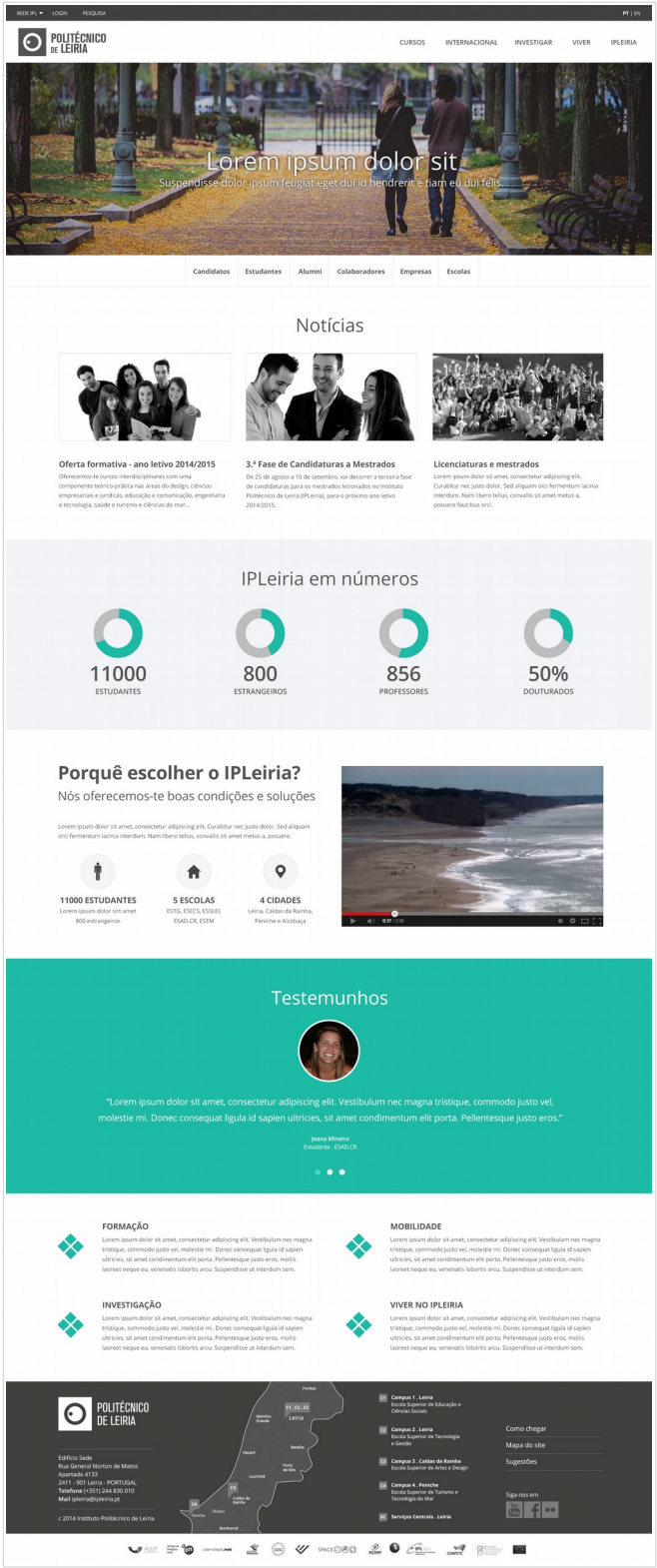
Anexo XVI - Proposta gráfica 5

SGCPI


POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
Lorem ipsum dolor sit
Suspendisse dolor ipsum feugiat eget dui id hendrerit e tiam eu dui felis.
Candidatos
Estudantes
Alumni
Docentes
Funcionários
Empresas
Notícias
Oferta formativa - ano letivo 2014/2015
3.ª Fase de Candidaturas a Mestrados
Licenciaturas e mestrados
IPLeiria em números
11000
ESTUDANTES
800
ESTRANGEIROS
856
PROFESSORES
50%
DOUTURADOS
Porquê escolher o IPLeiria?
Nós oferecemos-te boas condições e soluções
11000 ESTUDANTES
5 ESCOLAS
4 CIDADES
Testemunhos
"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum nec magna tristique, commodo justo vel, molestie mi. Donec consequat ligula id sapien ultricies, sit amet condimentum elit porta. Pellentesque justo eros."
Joana Mineiro
Estudante . ESAD.CR
FORMAÇÃO
MOBILIDADE
INVESTIGAÇÃO
VIVER NO IPLEIRIA
POLITÉCNICO DE LEIRIA
Edifício Sede
Rua General Norton de Matos
Apartado 4133
2411 - 901 Leiria - PORTUGAL
Telefone (+351) 244 830 010
Mail ipleiria@ipleiria.pt
c 2014 Instituto Politécnico de Leiria
Campus 1 . Leiria
Escola Superior de Educação e Ciências Sociais
Campus 2 . Leiria
Escola Superior de Tecnologia e Gestão
Campus 3 . Caldas da Rainha
Escola Superior de Artes e Design
Campus 4 . Peniche
Escola Superior de Turismo e Tecnologia do Mar
Serviços Centrais . Leiria
Como chegar
Mapa do site
Sugestões
Siga-nos em

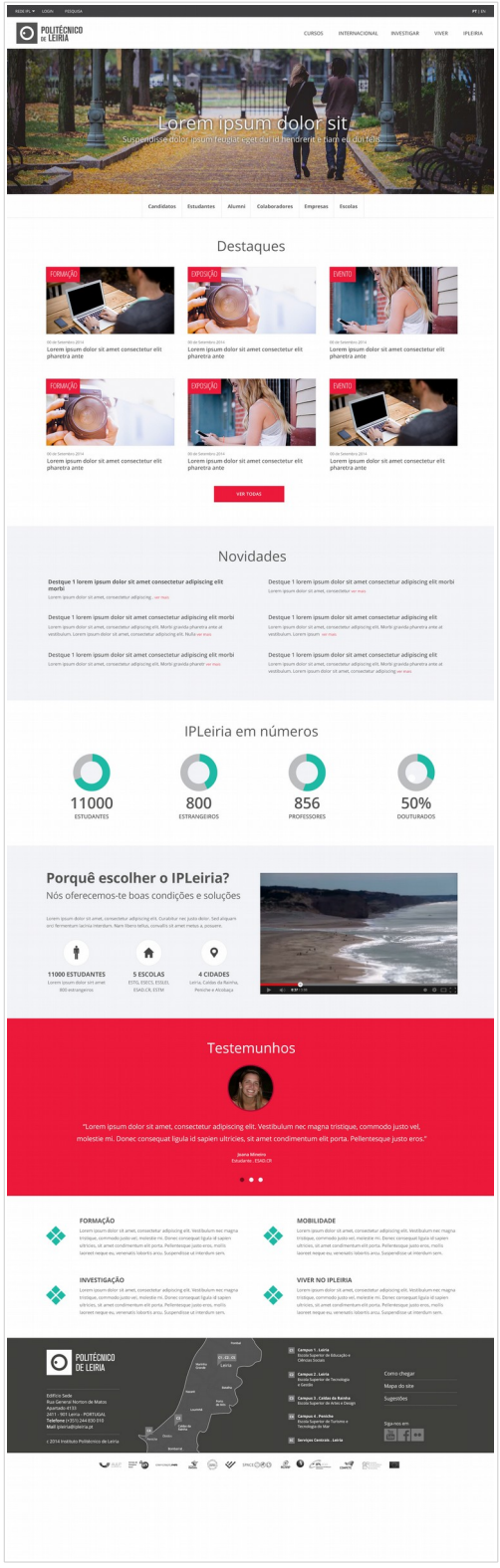
Anexo XVII - Proposta gráfica 6

SGCPI


POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
Lorem ipsum dolor sit
Suspendisse dolor ipsum feugiat eget dui id hendrerit etiam eu dui felis
Candidatos
Estudantes
Alumni
Colaboradores
Empresas
Escolas
Notícias
Oferta formativa - ano letivo 2014/2015
3.ª Fase de Candidaturas a Mestrados
Licenciaturas e mestrados
IPLeiria em números
11000
ESTUDANTES
800
ESTRANGEIROS
856
PROFESSORES
50%
DOUTURADOS
Porquê escolher o IPLeiria?
Nós oferecemos-te boas condições e soluções
11000 ESTUDANTES
5 ESCOLAS
4 CIDADES
Testemunhos
FORMAÇÃO
MOBILIDADE
INVESTIGAÇÃO
VIVER NO IPLEIRIA
Como chegar
Mapa do site
Sugestões
Siga-nos em

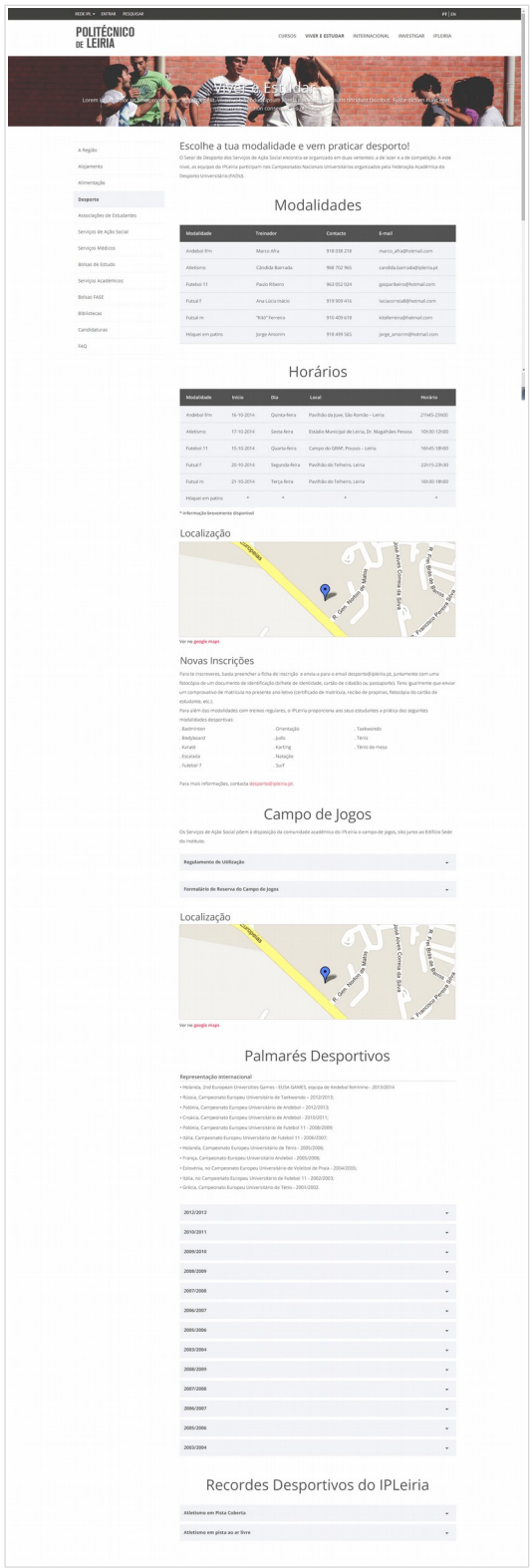
Anexo XVIII - Proposta gráfica 7

SGCPI


POLITÉCNICO DE LEIRIA
CURSOS
INTERNACIONAL
INVESTIGAR
VIVER
IPLEIRIA
Lorem ipsum dolor sit...
Candidatos
Estudantes
Alumni
Colaboradores
Empresas
Escolas
Destaques
FORMAÇÃO
EXPOSIÇÃO
EVENTO
VER TODAS
Novidades
IPLeiria em números
11000
ESTUDANTES
800
ESTRANGEIROS
856
PROFESSORES
50%
DOUTURADOS
Porquê escolher o IPLeiria?
Nós oferecemos-te boas condições e soluções
11000 ESTUDANTES
5 ESCOLAS
4 CIDADES
Testemunhos
FORMAÇÃO
MOBILIDADE
INVESTIGAÇÃO
VIVER NO IPLEIRIA
POLITÉCNICO DE LEIRIA
Como chegar
Mapa do site
Sugestões

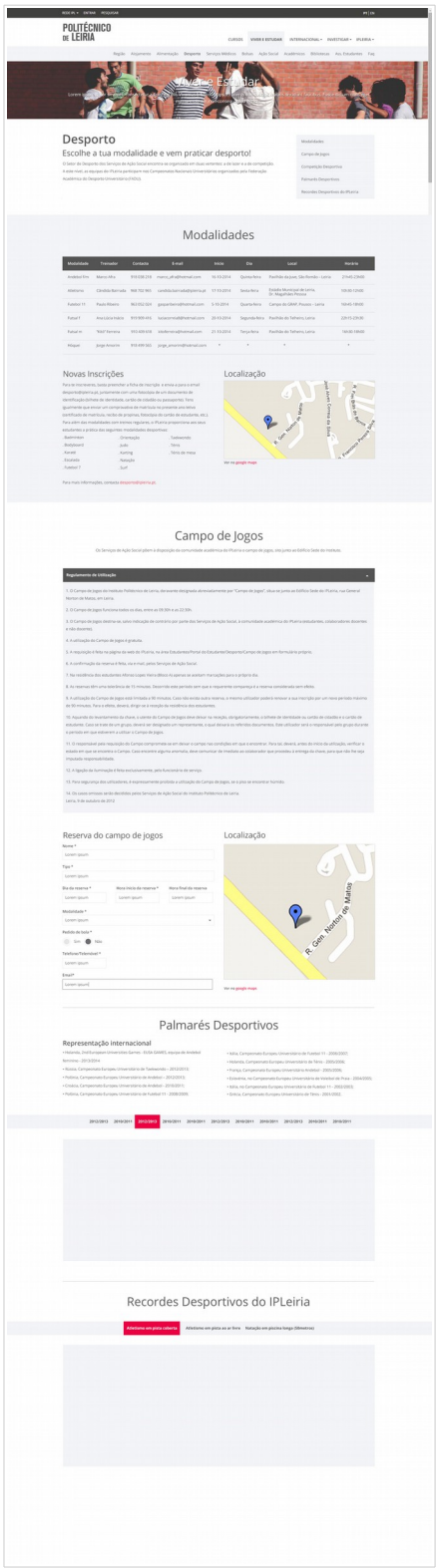
Anexo XIX - Proposta gráfica 1 para página com estrutura personalizada

SGCPI

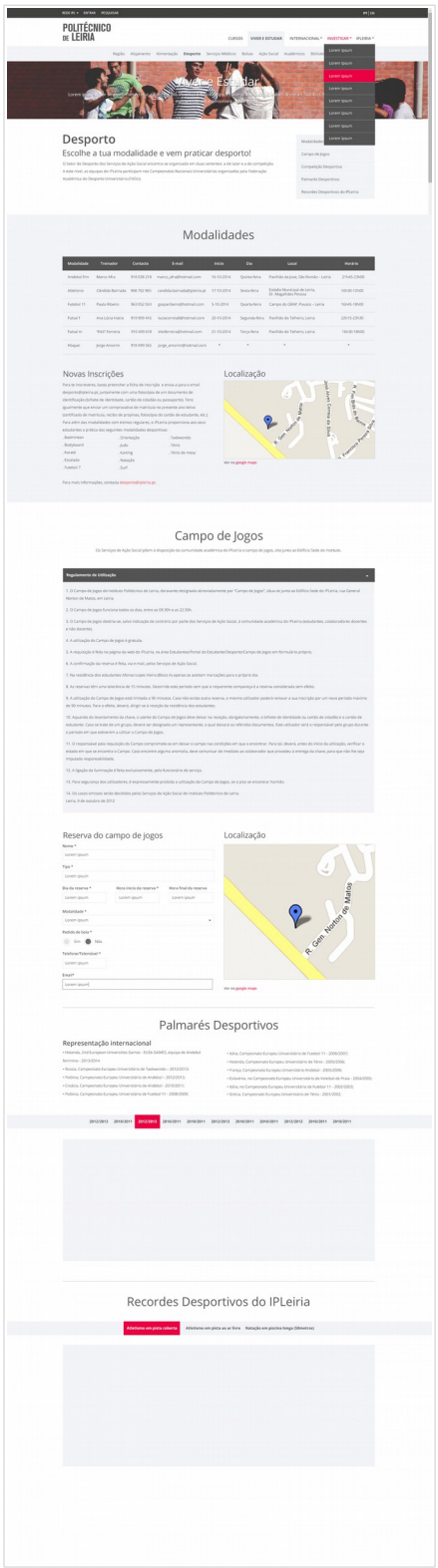

POLITÉCNICO DE LEIRIA
CURSOS
VIVER E ESTUDAR
INTERNACIONAL
INVESTIGAR
IPLEIRIA
Viver e Estudar
A Região
Alojamento
Alimentação
Desporto
Associações de Estudantes
Serviços de Ação Social
Serviços Médicos
Bolsas de Estudo
Serviços Académicos
Bolsas FASE
Bibliotecas
Candidaturas
FAQ
Escolhe a tua modalidade e vem praticar desporto!
Modalidades
Horários
Localização
Novas Inscrições
Campo de Jogos
Palmarés Desportivos
Recordes Desportivos do IPLeiria

Anexo XX - Proposta gráfica 2 para página com estrutura personalizada

SGCPI


POLITÉCNICO DE LEIRIA
Desporto
Escolhe a tua modalidade e vem praticar desporto!
Modalidades
Novas Inscrições
Localização
Campo de Jogos
Reserva do campo de jogos
Localização
Palmarés Desportivos
Representação internacional
Recordes Desportivos do IPLeiria

Anexo XXI - Proposta gráfica com exemplo do funcionamento do menu


POLITÉCNICO DE LEIRIA
Desporto
Escolhe a tua modalidade e vem praticar desporto!
Modalidades
Novas Inscrições
Localização
Campo de Jogos
Reserva do campo de jogos
Localização
Palmarés Desportivos
Representação internacional
Recordes Desportivos do IPLeiria

Anexo XXII - Imagem promocional de lançamento do novo *site*

SGCPI


UM NOVO DESIGN PARA UM NOVO ANO
A NEW DESIGN FOR A NEW YEAR
MAIS FLEXÍVEL
RESPONSIVE DESIGN
MAIS AMIGÁVEL
MORE USER-FRIENDLY
MAIS ...
MORE ...

Anexo XXIII - Conteúdo do ficheiro Gruntfile.js

SGCPI

```
module.exports = function (grunt) {
  // Project configuration.
  grunt.initConfig({
    // Metadata.
    pkg: grunt.file.readJSON('package.json'),
    banner: '/*! <%= pkg.title || pkg.name %> - v<%= pkg.version %> - ' +
      '<%= grunt.template.today("yyyy-mm-dd") %>\n' +
      '<%= pkg.homepage ? "* " + pkg.homepage + "\\n" : "" %>' +
      '* Copyright (c) <%= grunt.template.today("yyyy") %> <%= pkg.author %>;' +
      ' License <%= pkg.license %> */\n',
    // Task configuration.
    compass: {
      build: {
        options: {
          sassDir: 'css/sass',
          cssDir: 'css',
          outputStyle: 'compressed',
          environment: 'production'
        }
      },
      from_watch: {
        options: {
          sassDir: 'css/sass',
          cssDir: 'css',
          /*outputStyle: 'compressed',*/
          environment: 'development'
        }
      }
    },
    csso: {
      build: {
        options: {
          report: 'min'
        },
        files: {
          'css/editor-style.css':'css/editor-style.css',
          'css/theme.css':'css/theme.css'/*,
          '***.dev.css'*/
        }
      }
    },
    autoprefixer: {
      build: {
        files: {
          'css/editor-style.css':'css/editor-style.css',
          'css/theme.css':'css/theme.css'/*,
          '***.dev.css'*/
        }
      }
    },
    cssmin: {
      build: {
        expand: true,
        src: ['**/*.dev.css'],
        ext: '.css'
      }
    },
    concat: {
      options: {
```

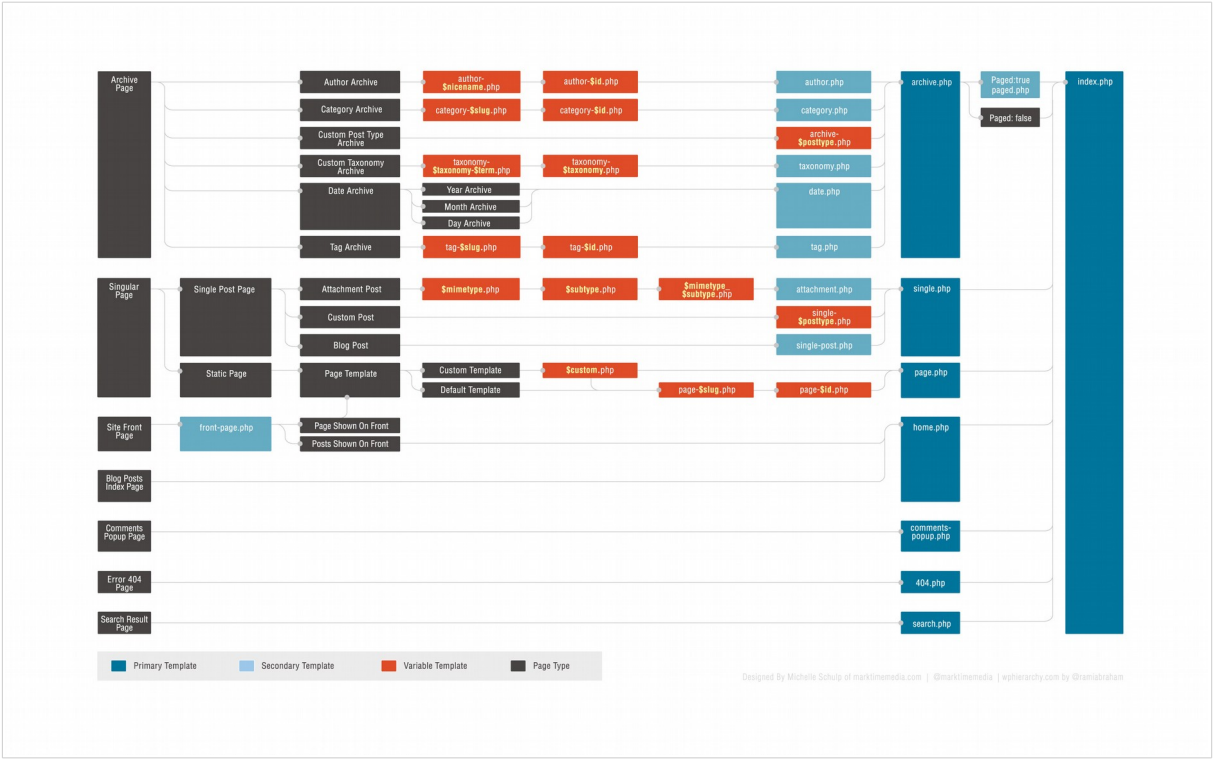
```
    stripBanners: false,
    separator: '\n'
  },
  build: {
    files: {
     'style.css': ['css/style.css', 'css/theme.css']
    }
  },
  from_watch: {
    files: {
     'style.css': ['css/style.css', 'css/theme.css']
    }
  }
},
uglify: {
  options: {
    banner: '<%= banner %>'
  },
  build: {
    expand: true,
    src: ['**/*.dev.js', '!*.min.js'],
    ext: '.min.js'
  },
  from_watch: {
    src: '*.dev.js',
    dest: '*.js'
  }
},
jshint: {
  options: {
    curly: true,
    unused: true,
    eqnull: true,
    latedef: true,
    newcap: true,
    nonew: true,
    trailing: true,
    evil: true,
    funcscope: true,
    noarg: true,
    sub: true,
    undef: true,
    boss: true,
    force: true,
    browser: true,
    globals: {
      jQuery: true
    }
  },
  build: {
    files: {
      src: ['**/*.dev.js']
    }
  },
  from_watch: {
    src: '*.dev.js',
    dest: '*.js'
  }
```

```
  },
  watch: {
    options: {
      nospawn: true
    },
    sass: {
      files: ['**/*.sass'],
      tasks: ['compass:from_watch','concat:from_watch']
    },
    scss: {
      files: ['**/*.scss'],
      tasks: ['compass:from_watch','concat:from_watch']
    },
    css: {
      files: ['**/*.dev.css'],
      tasks: ['cssmin:build']
    },
    js: {
      files: ['**/*.dev.js'],
      tasks: ['jshint:from_watch', 'uglify:from_watch']
    }/*,
    concat:{
      files: ['css/style.css','css/theme.css'],
      tasks: ['concat:from_watch']
    }*/
  },
  concurrent: {
    css: ['compass:build', 'autoprefixer:build', 'cssmin:build', 'concat:build'],
    js: ['jshint:build', 'uglify:build'],
    options: {
      limit: 5,
      logConcurrentOutput: true
    }
  }
});

grunt.event.on('watch', function (action, filepath, target) {
  if (target == "js") {
    var destFilePath = filepath.replace(/(.+)\.dev\.js$/, '$1.min.js');
    grunt.config('jshint.from_watch.src', filepath);
    grunt.config('uglify.from_watch.src', filepath);
    grunt.config('uglify.from_watch.dest', destFilePath);
  }
});

// These plugins provide necessary tasks.
grunt.loadNpmTasks('grunt-contrib-uglify');
grunt.loadNpmTasks('grunt-contrib-jshint');
grunt.loadNpmTasks('grunt-contrib-compass');
grunt.loadNpmTasks('grunt-contrib-cssmin');
grunt.loadNpmTasks('grunt-contrib-watch');
grunt.loadNpmTasks('grunt-concurrent');
grunt.loadNpmTasks('grunt-autoprefixer');
grunt.loadNpmTasks('grunt-csso');
grunt.loadNpmTasks('grunt-contrib-concat');

// Default task.
grunt.registerTask('default', ['watch']);
```

```
  //grunt.registerTask('build', ['concurrent:css', 'concurrent:js']);
  grunt.registerTask('build', ['compass:build', 'autoprefixer:build', 'csso:build',
'cssmin:build', 'concat:build','jshint:build', 'uglify:build']);
};
```

Anexo XXIV - Hierarquia dos modelos de um tema

SGCPI


Archive Page
Author Archive
Category Archive
Custom Post Type Archive
Custom Taxonomy Archive
Date Archive
Tag Archive
author-$nicename.php
category-$slug.php
taxonomy-$taxonomy-$term.php
Year Archive
Month Archive
Day Archive
tag-$slug.php
author-$id.php
category-$id.php
taxonomy-$taxonomy.php
tag-$id.php
author.php
category.php
archive-$posttype.php
taxonomy.php
date.php
tag.php
archive.php
Paged:true paged.php
Paged: false
index.php
Singular Page
Single Post Page
Static Page
Attachment Post
Custom Post
Blog Post
Page Template
Custom Template
Default Template
$mimetype.php
$subtype.php
$custom.php
$mimetype_$subtype.php
page-$slug.php
attachment.php
single-$posttype.php
single-post.php
page-$id.php
single.php
page.php
Site Front Page
front-page.php
Page Shown On Front
Posts Shown On Front
home.php
Blog Posts Index Page
Comments Popup Page
comments-popup.php
Error 404 Page
404.php
Search Result Page
search.php
Primary Template
Secondary Template
Variable Template
Page Type

Anexo XXV - Cabeçalho de um ficheiro gettext

```
msgid ""
msgstr ""
"Project-Id-Version: IPLeiria 2014.2\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2015-02-14 04:01-0000\n"
"PO-Revision-Date: \n"
"Last-Translator: Cláudio Esperança <cesperanc@gmail.com>\n"
"Language-Team: Cláudio Esperança <cesperanc@gmail.com>\n"
"Language: pt_PT\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"
"X-Poedit-SourceCharset: UTF-8\n"
"X-Poedit-KeywordsList: __;_e;esc_attr_e;esc_attr__;esc_html__;esc_html_e;"
"_n:1,2;_n_noop:1,2;_nx:1,2;_nx_noop:1,2;_x:1,2c;_ex:1,2c;esc_attr_x:1,2c\n"
"X-Poedit-Basepath: .\n"
"Plural-Forms: nplurals=2; plural=n != 1;\n"
"X-Generator: Poedit 1.5.4\n"
"X-Poedit-SearchPath-0: ..\n"

#: ../loop.php:12
#, php-format
msgid "Continue reading %s"
msgstr "Continuar a ler %s"
```

Anexo XXVI - Conteúdo do ficheiro .gitattributes

SGCPI

.gitattributes merge=ours
package.json merge=ours
README.md merge=ours
screenshot.jpg merge=ours
style.css merge=ours
langs/* merge=ours
.directory merge=ours

Anexo XXVII - Conteúdo do ficheiro .gitignore

SGCPI

/_tmp/
/.project
.sass-cache/*
/*/.sass-cache
node_modules/*
.htaccess
/*/.gitignore
.directory

Anexo XXVIII - Exemplo de conteúdo do ficheiro README.md

SGCPI

IPLeiria Courses WordPress theme
====

WordPress theme for the IPLeiria Courses website.

Instruções para desenvolvimento
==========

Cópia do tema

O tema deve ser copiado a partir do repositório associado a este projeto para uma pasta com o nome `ipleiria_courses` na pasta `[WP_ROOT]/wp-content/themes/`, onde `[WP_ROOT]` corresponde à pasta raiz da instalação do WordPress em uso.

Exemplo, assumindo o diretório atual como `[WP_ROOT]/wp-content/themes/`:

```
git clone ssh://git@redmine.ipleiria.pt/p0061/p0118/wordpress-theme-ipleiria_courses.git
```

Suporte ao desenvolvimento

Este tema foi desenvolvido com suporte ao [Grunt**]|], um ambiente de execução de tarefas baseado em JavaScript.**

Para utilizar este recurso, depois de estar instalado o suporte para NodeJS, executar o comando seguinte para instalar os módulos necessários:

```
npm install
```

Isto deverá instalar alguns dos módulos necessários para o Grunt de acordo com as definições do ficheiro `package.json`, que especifica a seguinte lista de dependências:
>
*** grunt - sistema de execução de tarefas**
*** grunt-contrib-jshint - módulo para análise da qualidade do código JavaScript**
*** grunt-contrib-watch - módulo para monitorização de alterações em ficheiros**
*** grunt-contrib-compass - módulo para processamento e compilação de instruções Sass**
*** grunt-contrib-cssmin - módulo para minificação de ficheiros CSS**
*** grunt-contrib-uglify - módulo para minificação de ficheiros JavaScript**
*** grunt-concurrent - módulo para execução de tarefas de forma concorrente**
*** grunt-autoprefixer - módulo para remoção/adição de prefixos CSS proprietários (consoante a compatibilidade com as instruções utilizadas)**
*** grunt-csso - módulo para otimização de instruções CSS (por exemplo, remoção de instruções repetidas)**

De um modo geral, o sistema implementado faz o processamento de ficheiros nas seguintes situações:
>
*** os ficheiros `*.sass` e `*.scss` nas pastas `css/sass` são processados e uma versão compilada dos mesmos é criada na pasta imediatamente anterior (`css`) com a extensão `.css`.**
*** os ficheiros `css/editor-style.css` e `css/theme.css` são adicionados prefixos, otimizações e são também minificados.**
*** os ficheiros `*.dev.css` são minificados para ficheiros com a extensão `.css`.**
*** os ficheiros `*.dev.js` são validados e minificados para ficheiros com a extensão `.min.js`.**

Foram registadas duas tarefas principais para o Grunt:

1. `grunt` sem parâmetros é ativado o modo *watch* onde as alterações a ficheiros após a execução do comando são monitorizadas e os respetivos ficheiros associados a essas alterações são recriados (em caso de necessidade).

2. `grunt build` é executada a compilação de todos os ficheiros com as otimizações finais de produção.

Organização do tema
==========

O tema disponível no repositório foi desenvolvido sob o WordPress 3.9.2 e apresenta uma arquitetura modular. Estes módulos estão disponíveis na pasta `thememodules` do tema e são carregados de forma automática pelo tema de acordo com as necessidades. Estes

módulos integram diretamente com a API do WordPress e funcionam como pequenas extensões ao tema.

Módulos disponíveis
----

1. **Bar - Personaliza o aspeto gráfico da barra de topo disponibilizada pelo WordPress**

2. **Branding - Personaliza a marca gráfica do tema, como por exemplo a imagem no formulário de autenticação ou o ícone de apresentação em alguns navegadores Web**

3. **ContentMenu - Permite inserção de menus da zona de conteúdo através do editor WYSIWYG e disponibiliza um botão neste editor para a inserção/gestão facilitada dos menus e respetivo aspeto gráfica. Do ponto de vista de código, reconhece um shortcode com um formato específico para a substituição pelo respetivo menu: `[content_menu menu="submenu-pt" container_class="nav noicon centered-menu vertical nocollapse"]` onde `menu` corresponde ao nome/id do menu a apresentar e `container_class` apresenta as classes a utilizar para a personalização gráfica ("nav" - aplica os estilos de navegação; "noicon" - remove os ícones do tipo de hiperligação; "vertical" - aplica os estilos base do menu do tema e define o formato vertical; "centered-menu" - deve ser aplicado em conjunto com o estilo anterior, removendo a disposição vertical e passando a centrar o menu numa única linha; "nocollapse" - remove o botão em formato de hambúrguer tradicional em dispositivos móveis). Os parâmetros aceites por este são os mesmos definidos dos da [função wp_nav_menu](http://codex.wordpress.org/Function_Reference/wp_nav_menu) (com exceção dos parâmetros `theme_location`, `echo`, `fallback_cb` e `walker`, sem representação neste módulo.**

4. **ContentTemplates - permite a definição e utilização de modelos globais ou a definição de modelos de conteúdo inicial por tipo de conteúdo. Disponibiliza uma ferramenta para inserção facilitada dos modelos disponíveis no editor WYSIWYG.**

5. **Cycle - Módulo responsável pela substituição de um shortcode num slideshow onde os elementos descendentes imediatos são utilizados como slides no slideshow. O módulos disponibiliza uma série de funcionalidades como por exemplo a pré-visualização do slideshow e edição inline na vista WYSIWYG. As opções suportadas são as mesmas que as disponibilizadas pela [API do Cyle 2][], com a diferença que no nome de cada opção o carater "-" deve ser substituído por um "_" (devido a limitações da API de shortcodes do WordPress).**

6. **Favicon - Permite a transferência do favicon para os navegadores que suportam esta funcionalidade, quando este é solicitado diretamente no endereço principal do site (situação que acontece com alguns navegadores web). Por exemplo, assumindo o endereço raiz `http://ipleiria.pt`, se um browser solicitar o endereço `http://ipleiria.pt/favivon.ico` será devolvido o ícone disponível na pasta do tema.**

7. **Groundwork - módulo responsável pela integração da [framework GroundworkCSS](http://groundworkcss.github.io/groundwork) no tema. Esta framework fornece um conjunto ferramentas e definições HTML, CSS e JavaScript que permitem criar sites responsivos, semânticamente válidos e acessíveis. Neste módulo foi feita a integração da framework com a API do WordPress, sendo que as alterações ao código base (que foi necessário adaptar) estão disponíveis no repositório https://github.com/cesperanc/groundwork. A escolha desta framework em substituição de outras mais conhecidas (como por exemplo o bootstrap), deveu-se essencialmente a questões relacionadas com a acessibilidade.**

8. **Hacks - A inclusão de pequenas modificações à API do WordPress é geralmente feita aqui. Por exemplo, se existir necessidade de correções gráficas, adicionar outros formatos de ficheiros permitidos (para além dos por omissão), etc, isto é feito neste módulo.**

9. **IPLeiriaFooter - Este módulo define o interface administrativo para a gestão do conteúdo do rodapé da página.**

10. **Menu - Define o menu principal do site bem como o carregamento dos *scripts* adicionais que sejam necessários.**

11. **MetaInformation - Define campos que permitem definir metadados do site, como por exemplo a descrição e/ou as palavras-chave.**

12. **Options - Módulo que define uma camada de abstração para a inserção de opções do tema a partir de outros módulos.**

13. **Language - Módulo responsável pela integração das funcionalidades de multi-idioma no tema, com recurso à extensão [Polylang][] que deve estar instalada e configurada no site para que possa ser utilizada pelo tema e respetivos módulos.**

14. **PostTypeContentFilter - Permite definir uma [WP_Query](http://codex.wordpress.org/Class_Reference/WP_Query) na área de conteúdo com a aplicação de um modelo específico para a apresentação dos resultados. Os parâmetros suportados são os seguintes:**
** * query (necessário) - define os parâmetros da query a executar tal como é especificado na [página de documentação da [WP_Query](http://codex.wordpress.org/Class_Reference/WP_Query). Este parâmetro apenas suporta o formato string, o que pode**

limitar a utilização de algumas opções mais avançadas (como disjunções).
  * template (necessário) - indicação do modelo a aplicar para a apresentação dos resultados da query especificada. Deve corresponder a um ficheiro disponível na raiz da pasta do tema onde, assumindo o valor `modelo_query` para este parâmetro, o nome do ficheiro correspondente deve ser `post-type-template-modelo_query.php`. No modelo a instância obtida da execução da query estará disponível na variável `$postTypeContentFilterLoop`
  * title (opcional) - define o título a partir do conteúdo. Este valor estará definido na variável `$postTypeContentFilterTitle` no modelo.
  > Exemplo do shortcode: `[PostTypeFilter query="post_type=featuredbanners&taxonomy=featuredbanners-categories&term=ued-pt&field=slug" template="modelo_query" title="titulo da query"]`;
  > Exemplo de um modelo: ```<?php
if(isset($postTypeContentFilterLoop)):
    if($postTypeContentFilterLoop->have_posts ()):
        ?>
            <section class="post-type-template projects-containner">
                <?php if(!empty($postTypeContentFilterTitle)): ?>
                    <header class="title">
                        <h2><?php echo($postTypeContentFilterTitle); ?></h2>
                    </header>
                <?php endif; ?>

                    <ul class="posts-row">
                        <?php
                            $count = 0;
                            while ($postTypeContentFilterLoop->have_posts()) : $postTypeContentFilterLoop->the_post();
                                $id = get_the_ID();
                                $title = get_the_title();
                                ?>
                                    <li id="<?php echo(get_post_type()); ?>-<?php echo($id); ?>" <?php post_class(array('post', 'project', 'expandable-content-grid-item')); ?>>
                                        <a class="url" title="<?php echo(esc_attr(sprintf(__("More information about '%s'", THEME_NAME), $title))); ?>" href="<?php the_permalink(); ?>">
                                            <?php echo($title); ?>

                                    <!-- #post-## -->
                                <?php
                                $count++;
                            endwhile;
                        ?>

            </section>
        <?php
    endif;
    wp_reset_postdata();
endif;```;

15. **ResponsiveImages** - Módulo responsável pela gestão das imagens de modo a apresentar a imagem mais adequada em função da área disponível no cliente que visita o site. Ao ser enviada uma imagem para o WordPress, este módulo solicita ao sistema que sejam geradas várias versões da imagem enviada com vários tamanhos que são depois utilizados em substituição da imagem inserida no conteúdo de modo a economizar largura de banda e aumentar o desempenho global do site. Este módulo adiciona ainda uma camada de compatibilidade com os navegadores web que não suportam imagens responsivas para garantir o funcionamento desta funcionalidade nestas situações.
16. **Robots** - Gera um ficheiro virtual `robots.txt` disponível na raiz do site com as informações de indexação para motores de pesquisa.
17. **SideBars** - Regista e define as zonas de widgets disponíveis no site.
18. **SiteMap** - Gera um mapa do site a partir do menu principal do site e que é apresentado para efeitos de acessibilidade.
19. **SiteMapXml** - Gera o ficheiro `sitemap.xml` com as informações dos conteúdos públicos disponíveis no site e que devem ser indexados pelo motor de pesquisa.
20. **SiteStructureInformation** - Permite a definição do conteúdo com informação sobre a estrutura do site, útil, por exemplo, para utilizadores com deficiências.
21. **TechnicalSheet** - Permite a apresentação da ficha técnica do site.

22. **TinyMCE - Módulo de suporte responsável para configuração do editor TinyMCE do WordPress e disponibilização de uma API que pode ser utilizada pelos restantes módulos para adicionar elementos a este editor.**
23. **Utilitities - Biblioteca de métodos auxiliares genéricos.**
24. **ViewportChecker - Inclusão do código JavaScript necessário para adição de classes a determinados elementos consoante a posição dos mesmos na área em apresentação no navegador web. Isto permite, por exemplo, criar animações quando é efetuado o deslocamento da página e esta atinge a posição de um determinado elemento. Para o efeito deve ser adicionada a classe CSS `animated` ao elemento ao qual se pretende adicionar esta funcionalidade e definir os seguintes atributos:**
** * data-animation (necessário) - define qual a classe que deve ser adicionada ao elemento quando é atingido o ponto pretendido.**
** * data-animation-offset (opcional, por omissão 100) - define o número de pixeis antes ou depois do elemento onde a classe deve ser adicionada.**
** * data-animation-repeat (opcional, por omissão falso) - define se a animação deve ser repetida caso a posição de animação seja atingida novamente.**
** * data-animation-callback (opcional, por omissão uma função vazia) - função JavaScript com a assinatura `function(elem, action)` que deve ser executada após a adição ou remoção da classe especificada ao elemento.**
25. **WpCleanup - Módulo responsável por remover elementos ou funcionalidades não necessárias do WordPress (por exemplo, suporte a comentários).**

[Polylang]: http://wordpress.org/plugins/polylang/
[API do Cyle 2]: http://jquery.malsup.com/cycle2/api/#options
[Grunt**]: http://gruntjs.com/**

Anexo XXIX - Processamento do conteúdo do README.md

README

Instruções para desenvolvimento

Cópia do tema

O tema deve ser copiado a partir do repositório associado a este projeto para uma pasta com o nome `ipleiria_courses` na pasta `[WP_ROOT]/wp-content/themes/`, onde `[WP_ROOT]` corresponde à pasta raiz da instalação do WordPress em uso.

Exemplo, assumindo o diretório atual como `[WP_ROOT]/wp-content/themes/`:

```
git clone ssh://git@redmine.ipleiria.pt/p0061/p0118/wordpress-theme-ipleiria_courses.git
```

Suporte ao desenvolvimento

Este tema foi desenvolvido com suporte ao **Grunt**, um ambiente de execução de tarefas baseado em JavaScript.

Para utilizar este recurso, depois de estar instalado o suporte para NodeJS, executar o comando seguinte para instalar os módulos necessários:

```
npm install
```

Isto deverá instalar alguns dos módulos necessários para o Grunt de acordo com as definições do ficheiro `package.json`, que especifica a seguinte lista de dependências:

> - *grunt - sistema de execução de tarefas*
> - *grunt-contrib-jshint - módulo para análise da qualidade do código JavaScript*
> - *grunt-contrib-watch - módulo para monitorização de alterações em ficheiros*
> - *grunt-contrib-compass - módulo para processamento e compilação de instruções Sass*
> - *grunt-contrib-cssmin - módulo para minificação de ficheiros CSS*
> - *grunt-contrib-uglify - módulo para minificação de ficheiros JavaScript*
> - *grunt-concurrent - módulo para execução de tarefas de forma concorrente*
> - *grunt-autoprefixer - módulo para remoção/adição de prefixos CSS proprietários (consoante a compatibilidade com as instruções utilizadas)*
> - *grunt-csso - módulo para otimização de instruções CSS (por exemplo, remoção de instruções repetidas)*

De um modo geral, o sistema implementado faz o processamento de ficheiros nas seguintes situações:

> - *os ficheiros `*.sass` e `*.scss` nas pastas `css/sass` são processados e uma versão compilada dos mesmos é criada na pasta imediatamente anterior (`css`) com a extensão `.css`.*
> - *os ficheiros `css/editor-style.css` e `css/theme.css` são adicionados prefixos, otimizações e são também minificados.*
> - *os ficheiros `*.dev.css` são minificados para ficheiros com a extensão `.css`.*
> - *os ficheiros `*.dev.js` são validados e minificados para ficheiros com a extensão `.min.js`.*

Foram registadas duas tarefas principais para o Grunt:

1. `grunt` sem parâmetros é ativado o modo *watch* onde as alterações a ficheiros após a execução do comando são monitorizadas e os respetivos ficheiros associados a essas alterações são recriados (em caso de necessidade).
2. `grunt build` é executada a compilação de todos os ficheiros com as otimizações finais de produção.

Organização do tema

O tema disponível no repositório foi desenvolvido sob o WordPress 3.9.2 e apresenta uma arquitetura modular. Estes módulos estão disponíveis na pasta `thememodules` do tema e são carregados de forma automática pelo tema de acordo com as necessidades. Estes módulos integram diretamente com a API do WordPress e funcionam como pequenas extensões ao tema.

Módulos disponíveis

1. **Bar** - Personaliza o aspeto gráfico da barra de topo disponibilizada pelo WordPress
2. **Branding** - Personaliza a marca gráfica do tema, como por exemplo a imagem no formulário de autenticação ou o ícone de apresentação em alguns navegadores Web

3. **ContentMenu** - Permite inserção de menus da zona de conteúdo através do editor WYSIWYG e disponibiliza um botão neste editor para a inserção/gestão facilitada dos menus e respetivo aspeto gráfica. Do ponto de vista de código, reconhece um shortcode com um formato específico para a substituição pelo respetivo menu: `[content_menu menu="submenu-pt" container_class="nav noicon centered-menu vertical nocollapse"]` onde `menu` corresponde ao nome/id do menu a apresentar e `container_class` apresenta as classes a utilizar para a personalização gráfica ("nav" - aplica os estilos de navegação; "noicon" - remove os ícones do tipo de hiperligação; "vertical" - aplica os estilos base do menu do tema e define o formato vertical; "centered-menu" - deve ser aplicado em conjunto com o estilo anterior, removendo a disposição vertical e passando a centrar o menu numa única linha; "nocollapse" - remove o botão em formato de hambúrguer tradicional em dispositivos móveis). Os parâmetros aceites por este são os mesmos definidos dos da função wp_nav_menu (com exceção dos parâmetros `theme_location`, `echo`, `fallback_cb` e `walker`, sem representação neste módulo.
4. **ContentTemplates** - permite a definição e utilização de modelos globais ou a definição de modelos de conteúdo inicial por tipo de conteúdo. Disponibiliza uma ferramenta para inserção facilitada dos modelos disponíveis no editor WYSIWYG.
5. **Cycle** - Módulo responsável pela substituição de um shortcode num slideshow onde os elementos descendentes imediatos são utilizados como slides no slideshow. O módulos disponibiliza uma série de funcionalidades como por exemplo a pré-visualização do slideshow e edição inline na vista WYSIWYG. As opções suportadas são as mesmas que as disponibilizadas pela API do Cyle 2, com a diferença que no nome de cada opção o carater "-" deve ser substituído por um "_" (devido a limitações da API de shortcodes do WordPress).
6. **Favicon** - Permite a transferência do favicon para os navegadores que suportam esta funcionalidade, quando este é solicitado diretamente no endereço principal do site (situação que acontece com alguns navegadores web). Por exemplo, assumindo o endereço raiz `http://ipleiria.pt`, se um browser solicitar o endereço `http://ipleiria.pt/favivon.ico` será devolvido o ícone disponível na pasta do tema.
7. **Groundwork** - módulo responsável pela integração da framework GroundworkCSS no tema. Esta framework fornece um conjunto ferramentas e definições HTML, CSS e JavaScript que permitem criar sites responsivos, semânticamente válidos e acessíveis. Neste módulo foi feita a integração da framework com a API do WordPress, sendo que as alterações ao código base (que foi necessário adaptar) estão disponíveis no repositório https://github.com/cesperanc/groundwork. A escolha desta framework em substituição de outras mais conhecidas (como por exemplo o bootstrap), deveu-se essencialmente a questões relacionadas com a acessibilidade.
8. **Hacks** - A inclusão de pequenas modificações à API do WordPress é geralmente feita aqui. Por exemplo, se existir necessidade de correções gráficas, adicionar outros formatos de ficheiros permitidos (para além dos por omissão), etc, isto é feito neste módulo.
9. **IPLeiriaFooter** - Este módulo define o interface administrativo para a gestão do conteúdo do rodapé da página.
10. **Menu** - Define o menu principal do site bem como o carregamento dos *scripts* adicionais que sejam necessários.
11. **MetaInformation** - Define campos que permitem definir metadados do site, como por exemplo a descrição e/ou as palavras-chave.
12. **Options** - Módulo que define uma camada de abstração para a inserção de opções do tema a partir de outros módulos.
13. **Language** - Módulo responsável pela integração das funcionalidades de multi-idioma no tema, com recurso à extensão Polylang que deve estar instalada e configurada no site para que possa ser utilizada pelo tema e respetivos módulos.
14. **PostTypeContentFilter** - Permite definir uma WP_Query na área de conteúdo com a aplicação de um modelo específico para a apresentação dos resultados. Os parâmetros suportados são os seguintes:
    - query (necessário) - define os parâmetros da query a executar tal como é especificado na [página de documentação da WP_Query. Este parâmetro apenas suporta o formato string, o que pode limitar a utilização de algumas opções mais avançadas (como disjunções).
    - template (necessário) - indicação do modelo a aplicar para a apresentação dos resultados da query especificada. Deve corresponder a um ficheiro disponível na raiz da pasta do tema onde, assumindo o valor `modelo_query` para este parâmetro, o nome do ficheiro correspondente deve ser `post-type-template-modelo_query.php`. No modelo a instância obtida da execução da query estará disponível na variável `$postTypeContentFilterLoop`
    - title (opcional) - define o título a partir do conteúdo. Este valor estará definido na variável `$postTypeContentFilterTitle` no modelo. > Exemplo do shortcode: `[PostTypeFilter query="post_type=featuredbanners&taxonomy=featuredbanners-categories&term=ued-pt&field=slug" template="modelo_query" title="titulo da query"]`; > Exemplo de um modelo: `<?php if(isset($postTypeContentFilterLoop)): if($postTypeContentFilterLoop->have_posts()): ?> <section class="post-type-template projects-containner"> <?php if(!empty($postTypeContentFilterTitle)): ?> <header class="title"> <h2><?php echo($postTypeContentFilterTitle); ?></h2> </header> <?php endif; ?> <div`

```
class="projects list expandable-content-grid"> <ul class="posts-row"> <?php $count = 0;
while ( $postTypeContentFilterLoop->have_posts() ) : $postTypeContentFilterLoop-
>the_post(); $id = get_the_ID(); $title = get_the_title(); ?> <li id="<?php
echo(get_post_type()); ?>-<?php echo($id); ?>" <?php post_class(array('post',
'project', 'expandable-content-grid-item')); ?>> <a class="url" title="<?php
echo(esc_attr(sprintf(__("More information about '%s'", THEME_NAME), $title))); ?>"
href="<?php the_permalink(); ?>"> <span class="project-title"><?php echo($title); ?>
</span> <span class="clearer"></span> </a> </li><!-- #post-## --> <?php $count++;
endwhile; ?> </ul> <div class="clearer"></div> </div> </section> <?php endif;
wp_reset_postdata(); endif;;
```

15. **ResponsiveImages** - Módulo responsável pela gestão das imagens de modo a apresentar a imagem mais adequada em função da área disponível no cliente que visita o site. Ao ser enviada uma imagem para o WordPress, este módulo solicita ao sistema que sejam geradas várias versões da imagem enviada com vários tamanhos que são depois utilizados em substituição da imagem inserida no conteúdo de modo a economizar largura de banda e aumentar o desempenho global do site. Este módulo adiciona ainda uma camada de compatibilidade com os navegadores web que não suportam imagens responsivas para garantir o funcionamento desta funcionalidade nestas situações.
16. **Robots** - Gera um ficheiro virtual `robots.txt` disponível na raiz do site com as informações de indexação para motores de pesquisa.
17. **SideBars** - Regista e define as zonas de widgets disponíveis no site.
18. **SiteMap** - Gera um mapa do site a partir do menu principal do site e que é apresentado para efeitos de acessibilidade.
19. **SiteMapXml** - Gera o ficheiro `sitemap.xml` com as informações dos conteúdos públicos disponíveis no site e que devem ser indexados pelo motor de pesquisa.
20. **SiteStructureInformation** - Permite a definição do conteúdo com informação sobre a estrutura do site, útil, por exemplo, para utilizadores com deficiências.
21. **TechnicalSheet** - Permite a apresentação da ficha técnica do site.
22. **TinyMCE** - Módulo de suporte responsável para configuração do editor TinyMCE do WordPress e disponibilização de uma API que pode ser utilizada pelos restantes módulos para adicionar elementos a este editor.
23. **Utilitities** - Biblioteca de métodos auxiliares genéricos.
24. **ViewportChecker** - Inclusão do código JavaScript necessário para adição de classes a determinados elementos consoante a posição dos mesmos na área em apresentação no navegador web. Isto permite, por exemplo, criar animações quando é efetuado o deslocamento da página e esta atinge a posição de um determinado elemento. Para o efeito deve ser adicionada a classe CSS `animated` ao elemento ao qual se pretende adicionar esta funcionalidade e definir os seguintes atributos:
    - data-animation (necessário) - define qual a classe que deve ser adicionada ao elemento quando é atingido o ponto pretendido.
    - data-animation-offset (opcional, por omissão 100) - define o número de pixeis antes ou depois do elemento onde a classe deve ser adicionada.
    - data-animation-repeat (opcional, por omissão falso) - define se a animação deve ser repetida caso a posição de animação seja atingida novamente.
    - data-animation-callback (opcional, por omissão uma função vazia) - função JavaScript com a assinatura `function(elem, action)` que deve ser executada após a adição ou remoção da classe especificada ao elemento.
25. **WpCleanup** - Módulo responsável por remover elementos ou funcionalidades não necessárias do WordPress (por exemplo, suporte a comentários).

Anexo XXX - Código-fonte do micro-núcleo do tema base em PHP

SGCPI

```php
<?php

/**
* The theme module functions file
*
* @package IPLeiria
* @subpackage theme
* @since IPLeiria 2014.0
* @author Cláudio Esperança
*/

namespace UED;

/**
* Taxonomy extra form fields trait
*/
if (!trait_exists(__NAMESPACE__ . '\\tThemeModule')):
  trait tThemeModule {
    protected static $_className = __CLASS__;
    protected static $_initDone = false;

  }
endif;

/**
* Interface to be implemented by theme modules
*/
if (!class_exists(__NAMESPACE__ . '\\ThemeModule')):
  abstract class ThemeModule {

    /**
     * @var <string> with the class name
     */
    private static $internalDebugEnabled = false;

    public static function init() {
      if(!isset(static::$_initDone)):
        return true;
      endif;
      // Internal flag to check if the init is done
      if (!static::$_initDone):
        static::$_initDone = true;
        return true;
      endif;
      return false;
    }

    /**
     * @uses has_filter() To verify if the filter was set.
     * @uses add_filter() To add the filter hook.
     */
    public static function addFilters($filters = array()) {
      if (isset(static::$_className) && is_array($filters) && !empty($filters)):
        foreach ($filters as $filter => $function):
          if (is_array($function)):
            foreach ($function as $functionName):
              if (has_filter($filter, array(static::$_className, $functionName)) === false):
                add_filter($filter, array(static::$_className, $functionName));
              endif;
            endforeach;
```

```
        else:
          if (has_filter($filter, array(static::$_className, $function)) === false):
            add_filter($filter, array(static::$_className, $function));
          endif;
        endif;
      endforeach;
    endif;
  }

  /**
   * @uses has_action() To verify if the action was set.
   * @uses add_action() To add the action hook.
   */
  public static function addActions($actions = array()) {
    if (isset(static::$_className) && is_array($actions) && !empty($actions)):
      foreach ($actions as $action => $function):
        if (is_array($function)):
          $functions = $function;
          foreach ($functions as $function):
            if (has_action($action, array(static::$_className, $function)) === false):
              add_action($action, array(static::$_className, $function));
            endif;
          endforeach;
        else:
          if (has_action($action, array(static::$_className, $function)) === false):
            add_action($action, array(static::$_className, $function));
          endif;
        endif;
      endforeach;
    endif;
  }

  /**
   * @return the prefix to append to the scripts and CSS files when the debug mode is enabled
   */
  public static function debugSufixJS() {
    return ((defined('SCRIPT_DEBUG') && SCRIPT_DEBUG) || self::$internalDebugEnabled ? '.dev' :
'.min');
  }

  /**
   * @return the prefix to append to the scripts and CSS files when the debug mode is enabled
   */
  public static function debugSufixCSS() {
    return ((defined('SCRIPT_DEBUG') && SCRIPT_DEBUG) || self::$internalDebugEnabled ? '.dev' :
'');
  }

  /**
   * Sets the post meta information
   * @param int $postId with the post identifier to associate the meta information with
   * @param string $key with the meta key name
   * @param string $value with the meta key value
   */
  public static function setPostMeta($postId = 0, $key = '', $value = '') {
    if (empty($key)):
      return false;
    endif;
    $postId = ( 0 === $postId ) ? get_the_ID() : $postId;
```

```
      if (get_post_meta($postId, $key, false)): //if the custom field already has a value
        update_post_meta($postId, $key, $value);
      else: //if the custom field doesn't have a value
        add_post_meta($postId, $key, $value);
      endif;
      if (empty($value)):
        delete_post_meta($postId, $key); //delete if any are blank, eg _dfcg-exclude is NULL
      endif;
    }

    /**
     * Retrieve the baseURI for the file
     *
     * @param string $file with the file fullpath
     * @param boolean $fromParentTheme load assets in relation to the parent theme instead of
the child theme. Defaults to true.
     * @return string with the URI
     *
     * @uses get_theme_root()
     * @uses get_stylesheet()
     * @uses get_bloginfo()
     */
    public static function getBaseUri($file = NULL, $fromParentTheme=true) {
      $themeModulesDirName = basename(dirname(__FILE__));
      $themeModulesDir = ($fromParentTheme?get_template():get_stylesheet()).DIRECTORY_SEPARATOR.
$themeModulesDirName.DIRECTORY_SEPARATOR;//dirname(__FILE__);
      $themeModulesUri = ($fromParentTheme?
get_template_directory_uri():get_stylesheet_directory_uri())."/{$themeModulesDirName}/";
      $moduleDir=dirname($file);
      $uri = $themeModulesUri.preg_replace('/.*'.preg_quote($themeModulesDir, '/').'(.*)/', '$1',
$moduleDir);

      return $uri;
    }

    /**
     * Retrieve the baseURI for the file in relation to a child theme
     *
     * @param string $file with the file fullpath
     * @return string with the URI
     */
    public static function getBaseUriFromChild($file = NULL) {
      return self::getBaseUri($file, (!is_child_theme()));
    }

    /**
     * Retrieve a post custom value
     *
     * @param string $key with the key name to get
     * @param int $post_id with the post id
     * @return mixed value or false
     *
     * @uses get_the_ID
     * @uses get_post_custom_values
     */
    public static function getPostSingleValue($key, $post_id = 0) {
      $post_id = ( 0 === $post_id ) ? get_the_ID() : $post_id;

      $value = get_post_custom_values($key, $post_id);
```

```
      if (!empty($value)):
        if (isset($value[0])):
          return maybe_unserialize($value[0]);
        endif;
        return maybe_unserialize($value);
      endif;
      return false;
    }

  }
endif;

/**
 * Base class for the theme with module support
 */
if (!class_exists(__NAMESPACE__ . '\\Theme')):
  class Theme extends ThemeModule {
    use tThemeModule;

    private static $instance = null;

    private function __construct() {
      //spl_autoload_extensions(spl_autoload_extensions().',.load.php');
      spl_autoload_register(array(__CLASS__, 'autoload'));
    }

    /**
     * Autoload static method for loading classes and interfaces.
     *
     * @param string $className The name of the class or interface.
     * @return void
     * @author Cláudio Esperança <claudio.esperanca@ipleiria.pt>
     */
    public static function autoload($className = '', $autoInit = false) {
      $classFile = preg_replace('/^' . preg_quote(__NAMESPACE__) . '(\:\:|\\\\)(.*)$/i', '$2',
$className);
      if (!empty($className)):
        $libraryPaths = explode(PATH_SEPARATOR, get_include_path());
        foreach ($libraryPaths as $libraryPath):
          $file = "{$libraryPath}{$classFile}" . DIRECTORY_SEPARATOR . "{$classFile}.load.php";
          if (is_readable($file) && strpos(dirname($file), get_theme_root()) === 0):
            include_once($file);

            // Lets verify if the class exists and implements the iThemeModule interface; if
not, continue the loading
            if ($autoInit && class_exists($className)):
              $oClass = new \ReflectionClass($className);
              if ($oClass->isSubclassOf(__NAMESPACE__ . '\\ThemeModule') && $oClass-
>hasMethod('init') && !static::$_initDone):
                $className::init();
                static::$_initDone = true;

                return;
              endif;
            endif;
          endif;
        endforeach;
      endif;
    }
```

SGCPI

```
    /**
     * Return the singleton instance
     *
     * @return <Theme> with the singleton instance
     */
    public static function getInstance() {
      if (empty(self::$instance)):
        self::$instance = new self();
      endif;
      return self::$instance;
    }

    /**
     * Add library paths to the include path
     *
     * @param <array> or <string> $libraryPath to add
     * @return <void>
     */
    public function addLibraryPaths($libraryPath = '') {
      if (empty($libraryPath)): // Empty library path, so cancel the operation
        return;
      elseif (is_array($libraryPath)): // If an array, remove all the prohibited items and
unlock the next step
        $allowedPath = false;
        $themeRoot = get_theme_root();
        foreach ($libraryPath as $key => $singleLibraryPath):
          if (!is_dir($singleLibraryPath) || strpos($singleLibraryPath, $themeRoot) !== 0):
            unset($libraryPath[$key]);
          elseif (!$allowedPath):
            $allowedPath = true;
          endif;
        endforeach;
      else: // If is a string and is in the allowed path, unlock the next step
        $allowedPath = (is_dir($libraryPath) && strpos($libraryPath, get_theme_root()) === 0);
      endif;

      if (!empty($libraryPath) && $allowedPath): // If we have items to add, verify such items
doesn't exist and add them
        $libraryPaths = explode(PATH_SEPARATOR, get_include_path());
        if (!in_array($libraryPath, $libraryPaths)):
          set_include_path((is_array($libraryPath) ? implode(PATH_SEPARATOR,
array_unique($libraryPath)) : $libraryPath) . PATH_SEPARATOR . get_include_path());
        endif;
      endif;
    }

    /**
     * Instanciates and load the class
     * @param <string> $className
     */
    public function loadClass($className = '') {
      if (!empty($className)):
        new $className();
      endif;
    }

    /**
     * Instanciates and loads the classes
     * @param <array> $classesName
```

```
*/
public function loadClasses($classesName = array()) {
  if (!empty($classesName) && is_array($classesName)):
    foreach ($classesName as $className):
      $this->loadClass($className);
    endforeach;
  endif;
}

/**
* Setup the theme
*/
public static function theme_setup() {
  load_theme_textdomain('IPLeiria', get_template_directory() . '/langs');

  if(is_child_theme()):
    load_child_theme_textdomain( 'IPLeiria', get_stylesheet_directory() . '/langs' );
  endif;

  add_theme_support('html5', array(
    'search-form',
    'gallery',
    'caption'
  ));
}

/**
* Load the theme main CSS file
*/
public static function wp_enqueue_scripts() {
  wp_enqueue_style(self::$_className, get_stylesheet_uri(), array(), '1.0.0');
}

/**
* Automagically loads the files to the theme
*
* @param string $dir, with the directory name to start the loading from
* @param [string $thatMatch], with the sufix to compare with the filename to load
* @param int $recursiveCount with the number of subdirectories to load
*
* @author Cláudio Esperança <claudio.esperanca@ipleiria.pt>
* @version 4.0
*/
public static function autoLoadFrom($dir, $thatMatch = '/.load.php$/', $recursiveCount = 0) {
  $dircontents = scandir($dir);
  foreach ($dircontents as $content):
    if ($content != "." && $content != ".."):
      if (preg_match($thatMatch, strtolower($content)) > 0 && is_file("{$dir}/{$content}")):
        self::getFile("{$dir}/{$content}");
      endif;
      if ($recursiveCount > 0 && is_dir("{$dir}/{$content}")):
        self::autoLoadFrom("{$dir}/{$content}", $thatMatch, $recursiveCount - 1);
      endif;
    endif;
  endforeach;
}

/**
* Load the file if it's on the theme dir
```

```
    * @param string $file with the filepath to load
    * @return boolean true on success, false otherwise
    *
    * @author Cláudio Esperança <claudio.esperanca@ipleiria.pt>
    * @version 3.0
    */
   public static function getFile($file) {
     if (is_readable($file) && stream_resolve_include_path($file) !== false/*&& dirname(__FILE__)
&& stripos(dirname(realpath($file)), dirname(__FILE__)) === 0*/):
       if (require_once($file)):
         return true;
       endif;
     endif;
     //error_log("Unable to load the file {$file}");

     return false;
   }

   /**
    * Theme initialization function
    */
   public static function init() {
     // Disable multiple initialization
     if(!parent::init()):
       return;
     endif;

     // Define the module files directory
     //self::getInstance()->addLibraryPaths(dirname(__FILE__).'/'); // doesn't work with
symlinks
     self::getInstance()->addLibraryPaths(get_template_directory() . DIRECTORY_SEPARATOR .
basename(dirname(__FILE__)) . DIRECTORY_SEPARATOR);

     // Load the modules
     self::autoLoadFrom(dirname(__FILE__) . DIRECTORY_SEPARATOR, '/.load.php$/', 1);

     // Register the theme setup hook
     self::addActions(array(
       'after_setup_theme' => 'theme_setup'
     ));
     // Add the default theme styleshet with very low priority to load after all the other
styles
     add_action('wp_enqueue_scripts', array(__CLASS__, 'wp_enqueue_scripts'), 99);

     remove_action('wp_head', 'wp_generator');
   }

 }
endif;

// Let's initialize the theme
Theme::init();
```

Anexo XXXI - Exemplo de implementação de um módulo em PHP

SGCPI

```
<?php
/**
 * Example module implementation
 *
 * @filesource thememodules/ExampleModule/ExampleModule.load.php
 * @package UED
 * @subpackage theme
 * @since UED 2015.0
 * @author Cláudio Esperança <cesperanc@gmail.com>
 */

// Encapsulate all the modules in a specific namespace to avoid collisions
namespace UED;

if(class_exists(__NAMESPACE__.'\\ThemeModule') && !
class_exists(__NAMESPACE__.'\\ExampleModule')):

    class ExampleModule extends ThemeModule{
        use tThemeModule;

        // Used to store the current module URL, to include other resources if needed
        private static $basedir = NULL;

        /**
         * Module initialization method
         */
        public static function init() {
            // Prevent multiple initialization
            if(!parent::init()):
                return;
            endif;

            // Store the current module URL
            self::$basedir = self::getBaseUri(__FILE__);

            // Bulk association of WordPress actions to module functions. Syntax:
'action_name'=>'class_method_name'
            // Standard add_action function from the WordPress API can be also used here
            self::addActions(array(
                'wp_enqueue_scripts'=>'wp_enqueue_scripts',
                //'admin_head'=>'admin_head',
                //'admin_enqueue_scripts'=>'admin_enqueue_scripts'
            ));
            // Bulk association of WordPress filters to module functions
            // Standard add_filter function from the WordPress API can be also used here
            self::addFilters(array(
                'the_content'=>'the_content',
                //self::FILTER_THEME_ADVANCED_STYLES=>'add_header_item'
            ));

            // We can also remove other filter or actions from other modules if needed
            //remove_filter( 'the_content', 'wpautop' );
        }

        /**
         * Adds a JavaScript library file to be loaded when a frontend is generated
         */
        public static function wp_enqueue_scripts(){
            // Retrieve the .dev or .min sufix to be added to the file name, depending on
whether we are in a development or production environment
            $suffix = self::debugSufixJS();
            wp_enqueue_script(__NAMESPACE__.'_ExampleJsLib', self::
$basedir."/js/jquery.examplemodule{$suffix}.js", array('jquery'), false, true);
        }
```

```
        /**
         * Filter the content of an article before presenting it to the user
         *
         * @param string $content
         * @return string The new content to be presented
         */
        public static function the_content($content){
            return $content." Hello World from the ExampleModule!";
        }
    }

    // Auto initialize the module
    ExampleModule::init();
endif;
```

Anexo XXXII - Código-fonte do ficheiro functions.php

```php
<?php
/**
 * The theme functions file
 *
 * @package IPLeiria
 * @subpackage theme
 * @since IPLeiria 2014.0
 * @author Cláudio Esperança
 */
require_once(dirname(__FILE__).'/thememodules/ThemeModules.php');
```

Anexo XXXIII - Código PHP do ficheiro index.php do tema base

```php
<?php
/**
 * The index file for the theme
 *
 * @package IPLeiria
 * @subpackage theme
 * @since IPLeiria 2014.0
 * @author Cláudio Esperança
 */
get_header();
  get_template_part('loop');
get_footer();
```

Anexo XXXIV - Código PHP do ficheiro loop.php do tema base

SGCPI

```
<?php
/**
 * The post loop
 *
 * @package IPLeiria
 * @subpackage theme
 * @since IPLeiria 2014.0
 * @author Cláudio Esperança
 */
if ( have_posts() ) :
    while (have_posts()) : the_post();
        $theContent = apply_filters('the_content', get_the_content(sprintf(__( 'Continue
reading %s', 'IPLeiria' ), '<span class="meta-nav">&rarr;</span>' )));
        $wrap = false;
        if(preg_match('/class=("|\')[^(\1)]*container[^(\1)]*(\1)/', $theContent)==0):
            $wrap = true;
            ?>
                <div class="container padded">
            <?php
        endif;
            ?>
                <div class="post <?php echo(is_single()?'single':'list'); ?>" id="post-<?php
the_ID(); ?>">
                    <?php if ( is_search() ) : // Only display excerpts for archives and
search. ?>
                        <<?php echo(is_single()?'h2':'h3'); ?> class="entry-title"><a href="<?
php the_permalink() ?>" rel="bookmark" title="<?php printf(__('Permanent Link to %s',
'IPLeiria'),the_title_attribute(array('echo'=>false))); ?>"><?php the_title(); ?></a></<?php
echo(is_single()?'h2':'h3'); ?>>
                        <div class="entry-summary">
                            <?php the_excerpt(); ?>
                        </div><!-- .entry-summary -->
                    <?php else :
                            edit_post_link(__('Edit', 'IPLeiria'), '<div class="container
align-right padded">', '</div>');
                    ?>
                        <div class="entry-content">
                            <div>
                                <?php echo($theContent); ?>
                            </div>
                            <div class="clearer"></div>
                        </div>
                    <?php endif; ?>
                </div>
            <?php
        if($wrap):
            ?>
                </div><!-- .container -->
            <?php
        endif;
    endwhile;
endif;
```

Anexo XXXV - Código PHP do micro-núcleo de um tema específico

```php
<?php

/**
 * The theme module functions file
 *
 * @package IPLeiria
 * @subpackage theme
 * @since IPLeiria 2014.0
 * @author Cláudio Esperança
 */

namespace UED;

// Let's initialize the theme
require_once( get_template_directory() . DIRECTORY_SEPARATOR . basename(dirname(__FILE__)) .
DIRECTORY_SEPARATOR . basename( __FILE__ ) );
if(class_exists(__NAMESPACE__ . '\\Theme')):
  $baseDir = get_stylesheet_directory() . DIRECTORY_SEPARATOR . basename(dirname(__FILE__)) .
DIRECTORY_SEPARATOR;
  Theme::getInstance()->addLibraryPaths($baseDir);
  Theme::autoLoadFrom($baseDir, '/.load.php$/', 1);
endif;
```

Anexo XXXVI - Código HTML para apresentação da Rede IPLeiria em *sites* externos

SGCPI

```
<!DOCTYPE html>
<html>
 <head>
  <title>Demo</title>
 </head>
 <body>
    <!-- base libraries -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <script type="text/javascript" src="http://www.ipleiria.pt/wp-
content/themes/ipleiria/thememodules/IFrameResizer/js/jquery-
responsiveiframe.min.js"></script>

    <!-- Toggle element; You can put it anywhere -->
  <a id="ipleiria-bar" href="http://www.ipleiria.pt/IPLeiriaBar.html?lang=en"
target="_blank">Network</a>

  <!-- IFrame to receive the content -->
   <iframe id="ipleiria-bar-panel" title="Rede IPLeiria"
src="http://www.ipleiria.pt/IPLeiriaBar.html?lang=en" class="IPLeiriaBarFrame" style="width:
100%; height: 0; display: none;">Loading...</iframe>

  <!-- Initialize the elements -->
    <script>
     (function ($) {
         if($('#ipleiria-bar-panel').responsiveIframe){
             $('#ipleiria-bar-panel').responsiveIframe({'xdomain': '*', 'autoHeight': false});
         }
         $("#ipleiria-bar").click(function(){
             $("#ipleiria-bar-panel").trigger("ResponsiveIframeToggle");
             return false;
         });
     })(jQuery.noConflict());
    </script>
 </body>
</html>
```

Anexo XXXVII - Código HTML para apresentação do rodapé do IPLeiria em *sites* externos

SGCPI

```
<!DOCTYPE html>
<html>
 <head>
  <title>Demo</title>
 </head>
 <body>
    <!-- base libraries -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <script type="text/javascript" src="http://www.ipleiria.pt/wp-
content/themes/ipleiria/thememodules/IFrameResizer/js/jquery-
responsiveiframe.min.js"></script>

  <!-- IFrame to receive the content -->
   <iframe id="ipleiria-footer-panel" title="Footer"
src="http://www.ipleiria.pt/IPLeiriaFooter.html?lang=en&amp;technical-sheet-url=false"
class="IPLeiriaFooterClientFrame" style="width: 100%; height: 0; border: 0;" sandbox="allow-
same-origin allow-top-navigation allow-forms allow-scripts">Loading...</iframe>

  <!-- Initialize the elements -->
    <script>
     (function ($) {
         if ($('#ipleiria-footer-panel').responsiveIframe) {
             $('#ipleiria-footer-panel').responsiveIframe({ 'xdomain': '*' });
         }
     })(jQuery.noConflict());
    </script>
 </body>
</html>
```

Anexo XXXVIII - Código PHP de um modelo do módulo PostTypeContentFilter

SGCPI

```
<?php
//[PostTypeFilter query="post_type=post&amp;post__in={sticky_posts}" template="posts"
defer_query="true" no_wrapper="true"]
if ($postTypeContentFilterQueryDefer && !empty($postTypeContentFilterQuery)):
    $query = wp_parse_args($postTypeContentFilterQuery);
    $supportsSticky = array(
        'post_parent__in',
        'post_parent__not_in',
        'post__in',
        'post__not_in'
    );
    foreach ($supportsSticky as $support):
        if (isset($query[$support]) && strtolower($query[$support]) == "{sticky_posts}"):
            $query[$support] = get_option('sticky_posts');
        endif;
    endforeach;
    $postTypeContentFilterLoop = new \WP_Query($query);
endif;
if (isset($postTypeContentFilterLoop) && is_object($postTypeContentFilterLoop) &&
is_a($postTypeContentFilterLoop, 'WP_Query')):

    if ($postTypeContentFilterLoop->have_posts()):
        wp_reset_query();
        $num_posts = $postTypeContentFilterLoop->found_posts;
        $posts_per_row = 3;
        $num_lines = floor($num_posts / $posts_per_row);
        if ($num_lines <= 1):
            $num_lines = 1;
        endif;
        $i = 0;

        $supports_post_thumbnails = current_theme_supports('post-thumbnails');
        ?>
        <div class="featured row">
            <?php
            while ($postTypeContentFilterLoop->have_posts()): $postTypeContentFilterLoop-
>the_post();
                $id = get_the_ID();
                //$category = get_the_category();
                $src = false;
                $width = 360;
                $height = 200;
                if ($supports_post_thumbnails && has_post_thumbnail($id) && ( $image =
wp_get_attachment_image_src(get_post_thumbnail_id($id))) :
                    list($src, $width, $height) = $image;
                endif;
                ?>

                <?php if (($i % $posts_per_row) == 0): ?><div class="row"><?php endif; ?>
                    <div class="one third double-padded" id="post-<?php the_ID(); ?>">
                        <a href="<?php echo get_permalink(); ?>">
                            <span class="post-image">
                                <!--<span class="post-category"><?php //echo $category[0]-
>cat_name;  ?></span>-->
                                <?php
                                if (!empty($src)):
                                    ?><span><img src="<?php echo($src); ?>" style="width: <?
php echo($width); ?>px" alt="<?php the_title(); ?>"/></span>
                                <?php else:
                                    ?><span class="image without-thumbnail"></span><?php
                                    endif;
                                    ?>
                            </span>
                            <span class="date"><?php echo get_the_date('j F Y'); ?></span>
```

```
                        <span class="title"><?php the_title(); ?></span>
                    </a>
                </div>
                <?php $i++;
                if (($i > 0) && ($i % $posts_per_row) == 0) || $i == $num_posts): ?>
                </div>
                <?php
                if ($num_lines <= 0):
                    break;
                endif;

            endif;
            ?>
        <?php endwhile; ?>
        </div>
        <?php
    endif;
    wp_reset_postdata();
endif;
```

Anexo XXXIX - Código PHP de um modelo para PostTaxonomyContentFilter

SGCPI

```
<?php
//[PostTaxonomyFilter taxonomies="course-category-type" template="courses-types"
no_wrapper="true"]

if($postTaxonomyContentFilterQueryDefer && !empty($postTaxonomyContentFilterTaxonomies)):
  $postTaxonomyContentFilterTerms = get_terms($postTaxonomyContentFilterTaxonomies,
$postTaxonomyContentFilterAttributes);
endif;

if(!empty($postTaxonomyContentFilterTerms)):
  if(!$postTaxonomyContentFilterNoWrapper):
  ?>
    <div class="post-taxonomy-template courses-types section">
      <?php if(!empty($postTaxonomyContentFilterTitle)): ?>
        <header class="title">
          <h2><?php echo($postTaxonomyContentFilterTitle); ?></h2>
        </header>
      <?php endif; ?>
      <div class="taxonomy list row">
        <?php
  endif;
          $thumbnailKey = class_exists("UED\CoursesCategoryArea")?
UED\CoursesCategoryArea::THUMBNAIL:'';
          $supportThumbnails = current_theme_supports('post-thumbnails');
          foreach ($postTaxonomyContentFilterTerms as $term):
            $thumbnail = null;
            if($supportThumbnails && class_exists("UED\CoursesCategoryArea") && !empty($term-
>$thumbnailKey)):
              $thumbnail = wp_get_attachment_image_src( $term->$thumbnailKey , 'full' , false );
            endif;
            $termLink = get_term_link( $term );
            if ( is_wp_error( $termLink ) ):
              continue;
            endif;
            ?>
              <article class="one fourth">
                <a class="link" href="<?php echo(esc_url( $termLink )); ?>">
                  <?php
                    if(!empty($thumbnail)):
                      list($src, $width, $height) = $thumbnail;
                      ?><div class="thumbnail" style="min-width: <?php echo($width); ?>px; min-
height: <?php echo($height); ?>px; background: url('<?php echo($src); ?>') no-repeat center
center;"></div><?php
                    endif;
                  ?>
                  <h3 class="title"><?php echo($term->name); ?></h3>
                  <p class="description" data-truncate="2"><?php echo($term->description); ?></p>
                </a>
              </article>
            <?php
          endforeach;

  if(!$postTaxonomyContentFilterNoWrapper):
        ?>
      </div>
    </div>
  <?php
  endif;
endif;
```

Anexo XL - Código HTML de uma ficha técnica

```
<div class="container padded">
<div class="row">
<h2>Ficha técnica</h2>
</div>
<div class="row">
<h3 class="header-item" style="font-weight: 600; margin-bottom: 0; padding: 10px; background-color: #545454; color: #fff;">Equipa</h3>
</div>
<div class="row padded" style="padding-top: 0;">[site_technical_sheet_theme_info wrap_with="&lt;div class="one sixth three-up-ipad two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Coordenação&lt;/h4&gt;&lt;ul&gt;%s&lt;/ul&gt;&lt;/div&gt;" row_format="&lt;li&gt;%s&lt;/li&gt;" retrieve_tag="UED Theme Coordinator"/]
<div class="one sixth three-up-ipad two-up-small-tablet" style="padding-right: 5px;">[site_technical_sheet_theme_info wrap_with="&lt;h4&gt;Design&lt;/h4&gt;&lt;ul&gt;%s&lt;/ul&gt;" row_format="&lt;li&gt;%s&lt;/li&gt;" retrieve_tag="UED Theme Designer"/]
<h4>Fotografia</h4>
<ul>
<li><a href="marcos.paixao@ipleiria.pt">Marcos Paixão</a></li>
</ul>
</div>
[site_technical_sheet_theme_info wrap_with="&lt;div class="one sixth three-up-ipad two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Programação&lt;/h4&gt;&lt;ul&gt;%s&lt;/ul&gt;&lt;/div&gt;" row_format="&lt;li&gt;%s&lt;/li&gt;" retrieve_tag="UED Theme Developer"/] [site_technical_sheet_theme_info wrap_with="&lt;div class="one sixth three-up-ipad two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Infraestrutura&lt;/h4&gt;&lt;ul&gt;%s&lt;/ul&gt;&lt;/div&gt;" row_format="&lt;li&gt;%s&lt;/li&gt;"]<a href="mailto:claudio.esperanca@ipleiria.pt" target="_blank">Cláudio Esperança</a>, <a href="mailto:nelson.matias@ipleiria.pt" target="_blank">Nelson Matias</a>, <a href="mailto:paulo.gomes@ipleiria.pt" target="_blank">Paulo Gomes</a>, <a href="mailto:ricardo.grilo@ipleiria.pt">Ricardo Grilo</a>, <a href="mailto:vitor.rodrigues@ipleiria.pt" target="_blank">Vitor Rodrigues</a>[/site_technical_sheet_theme_info] [site_technical_sheet_authors wrap_with="&lt;div class="one sixth three-up-ipad two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Conteúdos&lt;/h4&gt;&lt;ul&gt;%s&lt;/ul&gt;&lt;/div&gt;" row_format="&lt;li&gt;%s&lt;/li&gt;"][/site_technical_sheet_authors]
<div class="one sixth three-up-ipad two-up-small-tablet" style="padding-right: 5px;">
<h4>Consultadoria acessibilidade</h4>
<ul>
<li><a href="mailto:manuela.francisco@ipleiria.pt">Manuela Francisco</a></li>
<li>Norberto Sousa (<a href="http://www.comacesso.pt/" target="_blank">www.comAcesso.pt</a>)</li>
</ul>
</div>
</div>
<div class="row">
<h3 class="header-item" style="font-weight: 600; margin-bottom: 0; padding: 10px; background-color: #545454; color: #fff;">Informação de sistema</h3>
</div>
<div class="row padded" style="padding-top: 0;">[site_technical_sheet_system_os wrap_with="&lt;div class="one fourth two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Sistema operativo&lt;/h4&gt;%s&lt;/div&gt;"/] [site_technical_sheet_system_server wrap_with="&lt;div class="one fourth two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Servidor Web&lt;/h4&gt;%s&lt;/div&gt;"/] [site_technical_sheet_system_php wrap_with="&lt;div class="one fourth two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Versão de PHP&lt;/h4&gt;%s&lt;/div&gt;"/]
<div class="&quot;one">
<h4>Plataforma</h4>
<a href="http://wordpress.org" target="_blank">Wordpress</a></div>
</div>
<div class="row">
<h3 class="header-item" style="font-weight: 600; margin-bottom: 0; padding: 10px; background-color: #545454; color: #fff;">Informação sobre a plataforma</h3>
</div>
<div class="row padded" style="padding-top: 0;">[site_technical_sheet_system_platform_version wrap_with="&lt;div class="one sixth two-up-ipad two-up-small-tablet" style="padding-right: 5px;"&gt;&lt;h4&gt;Versão&lt;/h4&gt;%s&lt;/div&gt;"/] [site_technical_sheet_system_platform_theme wrap_with="&lt;div
```

class="one sixth two-up-ipad two-up-small-tablet" style="padding-right: 5px;"><h4>Tema</h4>%s</div>"/|
|site_technical_sheet_system_platform_plugins wrap_with="<div class="four sixth" style="padding-right:
5px;"><h4>Extensões</h4>%s</div>" row_format="%s, "/|
```
</div>
```

Anexo XLI - Modelo virtual utilizado na lista de cursos

```
<div class="drop-shadow"> </div>
<div class="container">
<div class="padded">
<div class="row align-center">
<h2 class="responsive" style="margin: 30px 0px;" data-compression="41" data-min="20" data-
scrollable="true">[ActionExecute action="courses-taxonomy-title" args="title=Cursos"]</h2>
</div>
</div>
<div class="row">
<aside class="one fifth padded right-four" style="padding-top: 0;">
<div>[ActionExecute action="type" args="title=Tipos"]</div>
<div>[ActionExecute action="school" args="title=Escolas Superiores"]</div>
<div>[ActionExecute action="area" args="title=Áreas"]</div>
<div>[ActionExecute action="period" args="title=Regimes"]</div>
<div>[ActionExecute action="clanguage" args="t<br> itle=Idiomas"]</div>
</aside>
<article class="four fifths left-one">
<div>[ActionExecute action="courses-taxonomy-description" wrap_with="&lt;div class='row padded'
style='padding: 0px 30px 30px 10px;'&gt;&lt;div class='box' style='padding: 30px;'&gt;
%s&lt;/div&gt;&lt;/div&gt;"]</div>
<div class="row">[PostTypeFilter template="courses"]</div>
</article>
</div>
</div>
```

Anexo XLII - Modelo virtual utilizado num item da lista de cursos

SGCPI

```
<h3 class="title">[PostCustomField field="post-title"]</h3>
<div class="meta">
<div class="">[PostCustomField field="school"]</div>
<div class="" style="border-top: 1px solid #FFFFFF; margin-top: 5px; padding-top: 5px;">Regime
[PostCustomField field="period"]</div>
</div>
```

Anexo XLIII - Modelo virtual utilizado na apresentação de um curso

```
<div class="drop-shadow"> </div>
<div class="container single courses" style="margin-bottom: 80px;">
<div class="padded">
<div><a class="[HistoryBackLinkClass]" style="display: none;" href="#">«
Voltar[HistoryBackLinkScript]</a></div>
<div class="row align-center">[PostCustomField field="post-title" wrap_with="&lt;h2&gt;
%s&lt;/h2&gt;"]</div>
</div>
<div class="row">
<article class="four fifths primary" style="padding-right: 25px;">
<div>[PostCustomField field="apresentacao-do-curso" wrap_with="&lt;div class='row
padded'&gt;&lt;div class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding:
10px;'&gt;&lt;h3 style='font-size: 1.3em;'&gt;Apresentação do curso&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="conteudo" wrap_with="&lt;div class='row padded'&gt;&lt;div
class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding:
10px;'&gt;&lt;h3&gt;Conteúdos&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="objectivos" wrap_with="&lt;div class='row padded'&gt;&lt;div
class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding:
10px;'&gt;&lt;h3&gt;Objetivos&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div class="row padded">[PostCustomField field="plano-curricular" wrap_with="&lt;div
class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding: 10px;'&gt;&lt;h3&gt;Plano
curricular&lt;/h3&gt;&lt;div &gt;%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="prosseguimento-de-estudos" wrap_with="&lt;div class='row
padded'&gt;&lt;div class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding:
10px;'&gt;&lt;h3&gt;Prosseguimento de estudos&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="condicoes-de-acesso" wrap_with="&lt;div class='row
padded'&gt;&lt;div class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding:
10px;'&gt;&lt;h3&gt;Condições de acesso&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="mais-informacoes" wrap_with="&lt;div class='row
padded'&gt;&lt;div class='box' style='margin-bottom: 35px;'&gt;&lt;div style='padding:
10px;'&gt;&lt;h3&gt;Mais informações&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]<br>[PostCustomField field="testemunhos"
wrap_with="&lt;div class='row padded'&gt;&lt;div class='box' style='margin-bottom:
35px;'&gt;&lt;div style='padding: 10px;'&gt;&lt;h3&gt;Testemunhos&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<p> </p>
<div></div>
</article>
<aside class="one fifth padded complementary">
<div>[PostCustomField field="codigo-do-curso" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Código curso&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="edital" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Edital&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="school" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Escola&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="clanguage" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Idioma&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="duracao" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Duração&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="period" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Regime&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="vagas" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Vagas&lt;/h3&gt;&lt;div &gt;
```

```
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="ects" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;ECTS&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="coordenador-de-curso" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Coordenador de curso&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="taxa-de-candidatura" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Taxa de candidatura&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="taxa-de-matricula" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Taxa de matrícula&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="propina" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Taxa anual&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="acreditacao" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Acreditação&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
<div>[PostCustomField field="registo-dges" wrap_with="&lt;div class='box'&gt;&lt;div
class='padded'&gt;&lt;h3&gt;Registo DGES&lt;/h3&gt;&lt;div &gt;
%s&lt;/div&gt;&lt;/div&gt;&lt;/div&gt;"]</div>
</aside>
</div>
</div>
```