

Benefícios e Desafios do Wordpress no Desenvolvimento de Sítios Web

Alexandre Miguel Leitão dos Santos

**Relatório de Estágio de Mestrado
em Novos Media e Práticas Web**

Fevereiro, 2015

Relatório de Estágio apresentado para cumprimento dos requisitos necessários à
obtenção do grau de Mestre em Novos Media e Práticas Web,
realizado sob a orientação científica do Professor Doutor Francisco Rui Cádima.

**BENEFÍCIOS E DESAFIOS DO WORDPRESS
NO DESENVOLVIMENTO DE SÍTIOS WEB
RELATÓRIO DE ESTÁGIO**

**BENEFITS AND CHALLENGES IN
WORDPRESS WEBSITE DEVELOPMENT
INTERNSHIP REPORT**

Alexandre Miguel Leitão dos Santos

RESUMO

O presente relatório foi elaborado no âmbito do estágio curricular do curso de mestrado em Novos Media e Práticas Web, da Faculdade de Ciências Sociais e Humanas da Universidade Nova de Lisboa. O estágio decorreu na Azorean – Aquatic Technologies, uma *spin-out* de uma conhecida empresa portuguesa do setor tecnológico, YDreams. O estágio teve como propósito geral refletir, através do trabalho efetuado na empresa, sobre os benefícios e os desafios da adoção do Wordpress no desenvolvimento de sítios Web. Quais são as vantagens do Wordpress em relação a outros sistemas de gestão de conteúdos? Que etapas compõem o desenvolvimento de um tema Wordpress? É preferível e mais eficaz começar o desenvolvimento de um *site* a partir de um tema *premium*, ou, pelo contrário, criar um tema de raiz? O objetivo específico do estágio e do presente relatório é responder a estas questões que se colocam a um Wordpress *developer*, tirando partido das dinâmicas práticas do estágio.

PALAVRAS-CHAVE: Desenvolvimento Web, Web design, linguagens Web, Wordpress.

ABSTRACT

This report was prepared under the internship of the master program in New Media and Web Practices, from the Faculty of Social and Human Sciences, New University of Lisbon. The internship took place in Azorean – Aquatic Technologies, a spin-out from a known Portuguese company in the technology sector, YDreams. As a general purpose, the goal of the internship was to think, through the work done in the company, about the benefits and challenges of using Wordpress in website development. What are the advantages of Wordpress compared to other content management systems? How to develop a Wordpress theme? It is better and more effective to start the development of a website from a premium theme or to build a Wordpress theme from scratch? The specific objective of this internship and report is to answer these questions, taking advantage of the dynamic internship practices.

KEYWORDS: Web development, Web design, Wordpress, Web practices.

ÍNDICE

Introdução	1
Capítulo I: Apresentação do estágio.....	3
I.1 A instituição de acolhimento	3
I.2 Objetivos do estágio.....	4
I.3 Plano de atividades	5
Capítulo II: Revisão da literatura	6
II.1 O Wordpress como CMS	6
II.2 Principais concorrentes.....	8
II.3 Temas Wordpress	10
II.4 Extensibilidade	12
II.5 Segurança.....	12
II.6 Usabilidade e acessibilidade	14
Capítulo III: Questões de ordem metodológica.....	17
Capítulo IV: Concretização do plano de atividades	21
IV.1 Produção e desenvolvimento Web.....	21
IV.2 Problemas encontrados na customização de alguns elementos	22
IV.3 Outras atividades desenvolvidas	24
Conclusão	26
Bibliografia.....	29
Webgrafia	32
Apêndice A: Protótipo “Azorean.eu”	a
Apêndice B: Protótipo “Myziphius.com”	b

Apêndice C: Página “Preview” da App Ziphius	c
Apêndice D: Relatório de acessibilidade “azorean.eu”	d
Apêndice E: Arte final “Azorean.eu”	e
Apêndice F: Arte final “Myziphius.com”	f
Apêndice G: Ficha da empresa	g

INTRODUÇÃO

O presente relatório foi elaborado no âmbito do estágio curricular do curso de mestrado em Novos Media e Práticas Web, da Faculdade de Ciências Sociais e Humanas da Universidade Nova de Lisboa. O estágio teve a duração total de 480 horas – cerca de três meses –, tendo decorrido entre 18 de agosto e 14 de novembro de 2014.

A minha grande motivação para este estágio curricular passava por praticar as linguagens Web que estudei durante a componente letiva do mestrado, nomeadamente nos Seminários de Práticas Web I e II. Sentia a necessidade urgente de treinar, de experimentar, impelido pela convicção aristotélica de que o que temos de aprender a fazer aprendemos fazendo. Porém, durante a minha entrevista de estágio na instituição de acolhimento, percebi que a Diretora de Operações e o Diretor de Design procuravam um estagiário com alguma experiência em Wordpress para desenvolver o *site* corporativo da empresa. O tema Wordpress já tinha sido adquirido, e necessitavam de alguém que percebesse, para além das linguagens Web convencionais, a linguagem mais específica desse gestor de conteúdos Web.

Para mim, a oportunidade de trabalhar com o Wordpress era ainda mais estimulante. Nos últimos anos, tenho-me dedicado a estudar essa plataforma como autodidata, e agora o estágio dava-me a hipótese de aprofundar ainda mais os meus conhecimentos, de maximizar e problematizar os meus processos de trabalho, aproveitando os noções de PHP¹ que tinha adquirido recentemente no curso de mestrado. Era um pretexto perfeito para pensar o Wordpress, o sistema de gestão de conteúdo Web com maior penetração no mercado.

Assim, este estágio tem como objetivos gerais refletir sobre o desenvolvimento de um produto Web baseado na plataforma Wordpress e enquadrar teoricamente o tema, mais particularmente sobre os benefícios e os desafios da adoção do Wordpress no desenvolvimento de sítios Web. Quais são as vantagens do Wordpress em relação a outros sistemas de gestão de conteúdos? Que etapas compõem o

¹ Sigla que significa Pré-Processador de Hipertexto. “O PHP é uma linguagem de programação para geração de documentos HTML. O PHP é interpretado e executado do lado do servidor, possibilitando o acesso a bases de dados” (Pereira e Poupa, 2011: 257).

desenvolvimento de um tema Wordpress? É preferível e mais eficaz começar o desenvolvimento de um *site* a partir de um tema Wordpress já existente, ou, pelo contrário, criar um tema de raiz? O objetivo específico do estágio e do presente relatório é, pois, responder a estas questões que se me colocam enquanto *Web developer*, tirando partido das dinâmicas práticas do estágio.

Do ponto de vista formal, o relatório está dividido em quatro capítulos: o primeiro faz uma breve apresentação do estágio – a instituição de acolhimento, os objetivos e o plano de atividades inicial; o segundo revê alguma literatura sobre o Wordpress e examina os aspetos mais importantes a considerar na criação e no desenvolvimento de um produto Web assente nessa plataforma; o terceiro incide nas questões metodológicas que se colocaram durante o estágio; e o quarto descreve as principais atividades executadas e problemas encontrados na customização de alguns elementos durante o trabalho de desenvolvimento Web. Por fim, a conclusão responde às questões de partida e propõe algumas recomendações para trabalhos futuros.

CAPÍTULO I: APRESENTAÇÃO DO ESTÁGIO

O estágio de natureza curricular, integrado no curso de mestrado em Novos Media e Práticas Web, da Faculdade de Ciências Sociais e Humanas da Universidade Nova de Lisboa, decorreu entre os dias 18 de agosto e 14 de novembro de 2014, e teve a duração total de 480 horas – cerca de três meses. Este capítulo apresenta a instituição de acolhimento, os objetivos do estágio e o plano de atividades previamente acordado com a instituição.

I.1 A instituição de acolhimento

A Azorean – Aquatic Technologies S.A. é uma *spin-out* da YDreams, uma conhecida empresa portuguesa especializada no desenvolvimento de experiências interativas através da combinação de tecnologia, arte e design, atualmente direcionada para as tecnologias no campo da realidade aumentada. A Azorean foi criada em 2012 e dedica-se ao desenvolvimento de produtos tecnológicos e instrumentos especializados de baixo custo destinados à exploração marinha. O objetivo da empresa é tornar-se pioneira na criação de sistemas robóticos autónomos e *gadgets* aquáticos, orientados a três diferentes segmentos de mercado: lazer, negócios e investigação científica.

O primeiro produto da Azorean chama-se Ziphius, um *drone* aquático controlado por *tablets* e *smartphones*, com data de comercialização prevista para junho de 2015. O produto foi o vencedor do concurso *Insert Coin: New Challengers*, promovido pela revista *online* de tecnologia Engadget (Santos, 2013), e foi inicialmente financiado por uma campanha bem sucedida do Kickstarter, através da qual a empresa angariou mais de 100 mil dólares (Azorean – Aquatic Technologies, 2013).

Em agosto de 2014, a Azorean procurava um *Web developer* com experiência em Wordpress para desenvolver um *site* corporativo, que, até então, se limitava a uma única página estática com os contactos da empresa. O carácter global do público-alvo da Azorean, a originalidade do produto que desenvolve e a necessidade de comunicar com os seus investidores tornavam urgente a criação de um *site* corpora-

tivo que pudesse ser uma montra da empresa para o mundo. Foi neste contexto que a minha colaboração com a Azorean se estabeleceu. A empresa manifestou-me a vontade de ter o *site* praticamente concluído em cerca de duas semanas, altura em que teria uma reunião importante com potenciais investidores. Foi esse carácter de urgência que deu consistência à hipótese de se trabalhar a partir do tema Wordpress adquirido pela empresa.

I.2 Objetivos do estágio

A minha grande motivação para este estágio passava por praticar as linguagens Web – especialmente HTML², CSS³ e PHP – que estudei durante a componente letiva do mestrado, nomeadamente nos Seminários de Práticas Web I e II. Sentia a necessidade urgente de treinar, de experimentar, impelido pela convicção aristotélica de que o que temos de aprender a fazer aprendemos fazendo, ou, como escreveu um dia o reputado designer italiano Bruno Munari, “a experimentação pessoal é que a melhor ensina” (2009: 18). Porém, durante a minha entrevista de estágio na instituição de acolhimento, percebi que a Diretora de Operações e o Diretor de Design procuravam um estagiário com alguma experiência em Wordpress para desenvolver o *site* corporativo da empresa. O tema Wordpress já tinha sido adquirido, e necessitavam de alguém que percebesse, para além das linguagens Web convencionais, a linguagem mais específica desse gestor de conteúdos Web.

A oportunidade de trabalhar com o Wordpress era, para mim, ainda mais estimulante e irresistível. Nos últimos anos, tenho-me dedicado a estudar essa plataforma como autodidata, e agora o estágio dava-me a hipótese de aprofundar ainda mais os meus conhecimentos, de maximizar e problematizar os meus processos de trabalho, aproveitando os noções de PHP que tinha adquirido recentemente no curso de mestrado. Era um pretexto perfeito para refletir sobre o Wordpress, o sistema de gestão de conteúdo Web com maior penetração no mercado.

² “O HTML é a linguagem básica da WWW. A maioria dos documentos na Internet encontra-se escrita em HTML, daí a sua incontornável importância” (Pereira e Poupa, 2008: 23).

³ As CSS (*Cascading Style Sheets* – Folhas de Estilo em Cascata) foram criadas com o objetivo “de separar a estrutura e o conteúdo de um documento (definidos em HTML) da sua apresentação (isto é, o aspeto e a formatação)” (Remoaldo, 2011: 5).

Assim, este estágio tem como objetivo geral analisar, através do trabalho efetuado na empresa, os benefícios e os desafios da adoção do Wordpress no desenvolvimento de sítios Web. Quais são as vantagens do Wordpress em relação a outros sistemas de gestão de conteúdos? Que etapas compõem o desenvolvimento de um tema Wordpress? Qual a melhor opção: começar o desenvolvimento de um *site* a partir de um tema Wordpress já existente, ou, pelo contrário, criar um tema de raiz (vantagens *versus* desvantagens)? O objetivo específico do estágio e do presente relatório é responder a estas questões que se me colocam enquanto *Web developer*, tirando partido das dinâmicas práticas do estágio.

I.3 Plano de atividades

O plano de atividades inicial não era muito detalhado: contemplava o desenvolvimento dos *sites* corporativos da Azorean e da YDreams, e a programação HTML/CSS da aplicação móvel do Ziphius. A empresa também manifestou interesse na hipótese, ainda pouco definida naquele momento, de eu poder executar alguns trabalhos de sonoplastia/sonorização, uma vez que já tinha experiência profissional nessa área. Em concreto, o plano de atividades consistia em três trabalhos relacionados com a programação de linguagens Web, dois deles em plataforma Wordpress:

1. Desenvolvimento do sítio Web corporativo da Azorean – Aquatic Technologies: pesquisa e estratégia; arquitetura da informação; análise de utilizadores; design de informação; design de interação; design sensorial; programação de linguagens Web; edição de vídeo; edição de imagem; implementação.
2. Desenvolvimento do sítio Web corporativo da YDreams: arquitetura da informação; design de informação; design de interação; design sensorial; programação de linguagens Web; edição de vídeo; edição de imagem; implementação.
3. Programação da Aplicação Móvel do Ziphius: programação de linguagens Web – HTML e CSS.

CAPÍTULO II: REVISÃO DA LITERATURA

Este capítulo apresenta os conceitos essenciais e faz uma pequena revisão da literatura sobre o trabalho desenvolvido durante o estágio. Como mais de 80 por cento do tempo foi passado a programar dois *sites* em plataforma Wordpress, uma grande parte da revisão da literatura incide, naturalmente, nessa temática mais específica. Apesar de ter feito um esforço para simplificar o mais possível a linguagem ao longo do relatório – mesmo correndo o risco de poder ser demasiado simplista –, reconheço que a linguagem utilizada neste capítulo é por vezes um pouco hermética. Mas, tratando este relatório de linguagens de programação tão específicas, seria difícil evitá-lo.

II.1 O Wordpress como CMS

O Wordpress é um conhecido sistema de gestão de conteúdo Web de código aberto, escrito em linguagem PHP e assente numa base de dados MySQL⁴. Lançado oficialmente a 27 de maio de 2003, por Matt Mullenweg e Mike Little, o Wordpress começou por ser mais um sistema de *blogging* do que um verdadeiro CMS (*Content Management System*). Algumas características que o tornaram popular continuam a aplicar-se nos dias de hoje: URLs⁵ “amigos” dos motores de busca, possibilidade de criação de múltiplas categorias, instalação simples, capacidade de *upgrade*, moderação de comentários e utilização e modificação completamente gratuitas, ao abrigo de uma licença GPLv2⁶. Rapidamente, o Wordpress tornou-se uma das plataformas mais requisitadas no mercado dos blogues, beneficiando da política de inflação de preços praticada pelo seu principal concorrente, Movable Type (Gavalda, 2014).

Porém, sobretudo a partir de 2010, com a introdução dos *custom post types* (tipos de conteúdo personalizado) na versão 3, aquilo que começou por ser uma

⁴ “O MySQL é um sistema de gestão de bases de dados relacional (SGBDR), ou seja, um sistema informático que permite armazenar dados de forma bem organizada e com baixa redundância, e possibilita a posterior manipulação dos dados armazenados” (Pereira e Poupa, 2011: 151).

⁵ Do inglês *Uniform Resource Locator*, traduzido para português como “Localizador Uniforme de Recurso”.

⁶ GPL (GNU *General Public Licence*) é a licença com maior utilização em projetos de *software* livre, sobretudo devido à sua adoção no projeto GNU e no sistema operativo GNU/Linux.

simples plataforma de administração de blogues evoluiu para um robusto sistema de gestão de conteúdos Web. Com um painel de administração intuitivo mas poderoso, a plataforma permite aquilo que qualquer CMS se propõe fazer: tornar mais fácil o controlo do conteúdo, ou seja, permitir que qualquer pessoa possa escrever, editar e publicar o seu trabalho, através de diversas ferramentas de fácil manuseamento – como, por exemplo, um editor de texto WYSIWYG (*What You See Is What You Get*) –, sem necessitar de ter um conhecimento aprofundado das linguagens Web.

Hoje, o mercado dos CMSs é enorme e a grande dificuldade é escolher. Não é possível dizer perentoriamente que há um melhor do que outros. E há soluções gratuitas e pagas, ambas com prós e contras. A opção por uma delas depende dos objetivos do projeto, do orçamento disponível, da preferência pessoal do(s) Web *developer(s)* e do grau de customização e de extensibilidade pretendidos.

Mas porventura aquilo que mais destaca o Wordpress da concorrência paga é o facto de ser completamente gratuito. Thord Hedengren (2010: 174), um experiente Web designer e *developer* especialista em Wordpress, gosta de dizer que todos os jornais e revistas *online* que conhece, e que estão assentes em sistemas com licenças de utilização caríssimas, poderiam facilmente ser reconstruídos em Wordpress, mantendo o mesmo desempenho. Sistemas como o Wordpress, o Joomla ou o Drupal – os três CMSs com maior penetração no mercado – têm não só a vantagem de ser gratuitos, como garantem uma expansibilidade maior e mais barata, ao permitirem a criação e o desenvolvimento de *plugins*, módulos e extensões por terceiros.

O facto de o Wordpress ter começado por ser uma plataforma de *blogging* explica, em grande medida, o sucesso que tem hoje enquanto CMS: muitos *bloggers* já familiarizados com o *back end*⁷ do sistema puderam passar a gerir um completo e poderoso gestor de conteúdos sem terem de aprender tudo de novo. Para estes utilizadores, o Wordpress proporcionava uma curva de aprendizagem muito mais rápida. Não é por acaso que o Wordpress é hoje, por uma esmagadora margem, o CMS mais popular, com cerca de 60 por cento de penetração no mercado (W3Techs, 2015); e 22

⁷ O *back end* de um *site* Wordpress é aqui entendido como o painel de administração, através do qual os utilizadores podem não só inserir e editar conteúdos, como também alterar funcionalidades, comportamentos e o design do *site*.

por cento dos *sites* em toda a Web já correm nesta plataforma (Gavalda, 2014). Entre os mais populares, encontram-se os *sites* *The New Yorker*, *TechCrunch*, *BBC America*, *Sony Music*, *Variety*, *MTV News*, *Lumia Conversations*, *Fortune*, *CNN Political Ticker*, *Reuters Blog* ou *Wired*, para citar apenas alguns. Estes dados não são despidiendos: como veremos mais adiante, a popularidade do Wordpress é a sua grande vantagem e, ao mesmo tempo, a sua maior fragilidade do ponto de vista da segurança.

Acrescentaria, como nota secundária e pessoal, que uma das razões que me levaram a simpatizar imediatamente com o Wordpress foi o facto de todas as versões terem nomes de músicos de jazz. Para um melómano como eu, seria difícil ser insensível a uma particularidade tão original.

II.2 Principais concorrentes

Wordpress, Joomla e Drupal são, por uma larga distância, os sistemas de gestão de conteúdos Web com maior penetração num mercado de centenas de CMSs, representando os três uma quota de mais de 70 por cento. Estes dados fazem do Joomla e do Drupal os maiores concorrentes do Wordpress. Contudo, as duas plataformas têm assistido a uma diminuição consecutiva da sua quota de mercado desde 2010: atualmente, o Joomla tem sete por cento de quota de mercado, contra 12 por cento em 2010; o Drupal tem apenas cinco, sendo que em 2010 tinha sete. O Wordpress, pelo contrário, tem hoje uma quota de 61 por cento, contra 51 em 2010 (W3Techs, 2015). Em apenas cinco anos, o Wordpress ampliou a sua quota de mercado em 10 pontos percentuais.

Apesar de algumas diferenças, é mais aquilo que os une do que os separa. Os três CMSs têm muitas características em comum e, com o tempo, as semelhanças entre eles têm vindo a aumentar consideravelmente. Por essa razão, não é possível dizer que um é superior aos outros. A opção por um deles depende sobretudo da preferência pessoal do(s) *Web developer(s)*. Entre os atributos comuns, ressaltam três óbvios: todos os CMS são *open source*, escritos em linguagem PHP e assentes numa base de dados MySQL. Um quadro comparativo com as principais características dos três gestores de conteúdos pode ser visto na página seguinte.

WORDPRESS	JOOMLA	DRUPAL
Lançado em 2003	Lançado em 2005	Lançado em 2001
TECNOLOGIA		
PHP MySQL	PHP MySQL	PHP MySQL
POPULARIDADE		
60,8% quota de mercado CMS 140 milhões de <i>downloads</i>	7,4% quota de mercado CMS 30 milhões de <i>downloads</i>	5,1% quota de mercado CMS 15 milhões de <i>downloads</i>
PREÇO		
Gratuito	Gratuito	Gratuito
ALGUNS SITES CONHECIDOS ASSENTES NA PLATAFORMA		
The New Yorker Sony Music BBC America Wired	Harvard University Linux ITWire MTV Greece	The White House Warner Bros. Records The Economist Twitter Developers
TEMAS GRATUITOS		
2000+	900+	1800+
PLUGINS GRATUITOS		
27000+	7000+	24000+
INSTALAÇÃO AUTOMÁTICA "NUM CLIQUE"		
Sim	Sim	Sim
DURAÇÃO DA INSTALAÇÃO MANUAL		
5 minutos	10 minutos	10 minutos
FACILIDADE DE MODERAÇÃO		
*	**	***
GRAU DE DOMÍNIO NECESSÁRIO DAS LINGUAGENS WEB		
*	**	***
FREQUÊNCIA DOS <i>UPDATES</i>		
42 dias	36 dias	51 dias
MELHOR UTILIZADO PARA...		
- Blogues - <i>Sites</i> corporativos - Sítios Web de tamanho pequeno/médio - Jornais/revistas <i>online</i>	- <i>Sites</i> de <i>e-commerce</i> - Sítios Web de <i>social networking</i>	- <i>Sites</i> de qualquer categoria - Sítios Web que exijam uma organização de dados mais complexa

Quadro 1 - Quadro comparativo dos três CMSs mais populares, a partir da informação recolhida de W3Techs (2015), Mening (2013), Powell (s.d.), Rackspace Support (2013) e Hedengren (2010).

Apesar das semelhanças, Mening (2013), Powell (s.d.), Rackspace Support (2013) e Hedengren (2010) são unânimes em destacar duas grandes diferenças entre o Wordpress e as plataformas concorrentes: por um lado, o Wordpress exige um grau de domínio inferior das linguagens Web, sendo um sistema mais fácil de compreender e de expandir a partir do painel de administração; por outro, a maior comunidade *online* de *developers* e de utilizadores garante ao Wordpress uma extensibilidade superior. No entanto, para alguém que domine as linguagens Web, estas diferenças são muito pouco significativas.

II.3 Temas Wordpress

Os temas são um dos elementos centrais do Wordpress. Na verdade, uma grande parte do trabalho de Web design de um *site* Wordpress resume-se ao desenvolvimento do tema. O tema consiste numa pasta com vários ficheiros essenciais para o funcionamento do *site*, que têm impacto tanto na forma (design) como na função (funcionalidades e customizações de comportamento através de código adicional). De todos os ficheiros, há apenas dois que são obrigatórios: `<style.css>` e `<index.php>`. O primeiro tem o cabeçalho que identifica o tema, o segundo é o *layout* básico do *site*. Embora fique um pouco limitado, é perfeitamente possível criar um *site* Wordpress apenas com estes dois ficheiros (Hedengren, 2010: 58).

Um aspeto importante a considerar no desenvolvimento de um tema é a hierarquia pela qual os ficheiros são evocados pelo sistema. O Wordpress está preparado para “chamar” certos ficheiros em função do seu nome. Por exemplo, quando no *site* abrimos um arquivo de uma categoria – no Wordpress, esse arquivo corresponde ao ficheiro `<category.php>` –, o sistema procura em primeiro lugar por um ficheiro com o nome dessa categoria. Imaginando que temos um *site* com a categoria “Ideias”, ao clicarmos nessa categoria o Wordpress procurará primeiro por um ficheiro chamado `<category-ideias.php>`. Se esse ficheiro não existir, a plataforma procura pelo número dessa categoria. Ou seja, se a categoria “Ideias” for a categoria 3, procurará por um ficheiro com o nome `<category-3.php>`. Se esse ficheiro também não existir, o sistema procurará pelo ficheiro mais genérico `<category.php>`. E

por aí em diante. Compreender esta hierarquia é fundamental para se tirar o máximo partido da customização da plataforma. A hierarquia completa pode ser consultada na página oficial do Wordpress (Wordpress Codex, 2015b).

Para quem esteja familiarizado com a linguagem PHP, a programação em Wordpress é simples e rápida de aprender. Há, no entanto, algumas nuances que devem ser consideradas. As boas práticas de programação em Wordpress aconselham a não utilizar algumas expressões PHP sempre que exista uma função Wordpress correspondente. Por exemplo, no desenvolvimento de um tema, as expressões `<include(>`, `<require(>`, `<include_once(>` e `<require_once(>` (Serrão e Marques, 2009: 88) devem ser preteridas a favor da função interna `<get_template_part(>`. É também desaconselhado embutir ficheiros CSS e Javascript⁸ diretamente a partir dos ficheiros que contêm o HTML, mas antes chamá-los através das funções internas `<wp_register_style(>`, `<wp_enqueue_style(>`, `<wp_register_script(>` e `<wp_enqueue_script(>`, a partir de ficheiros que alberguem todas as funções do tema. São apenas dois exemplos, entre muitos outros, de como as funções do Wordpress devem ser privilegiadas, de modo a garantir uma maior consistência, compatibilidade, flexibilidade e extensibilidade do código. A documentação é outra grande vantagem do Wordpress: todas as funções, *hooks*, classes e métodos estão disponíveis *online* para consulta imediata (Wordpress Codex, 2015a).

Nos últimos anos, tem crescido o mercado dos temas *premium*: temas Wordpress já desenhados e programados, criados por Web designers e *developers* profissionais, repletos de características avançadas e funções adicionais. Ao contrário da licença de utilização do Wordpress, os temas *premium* são pagos.

⁸ O Javascript é uma linguagem de programação da família da linguagem C. “Uma das principais características do Javascript é a forma como interage com o navegador, podendo aceder aos seus objetos” (Pereira e Poupa, 2008: 93).

II.4 Extensibilidade

Por defeito, o Wordpress é já uma plataforma bastante poderosa, que pode ser utilizada para desenvolver uma grande variedade de *sites* dinâmicos. Mas a expansibilidade do sistema, garantida através de temas, *plugins*, *scripts* e funções personalizadas, é uma das razões que explica a sua enorme popularidade (Coyier e Starr, 2012: 131). A popularidade do Wordpress contribuiu para a criação e o crescimento da maior comunidade *online* de *developers* e entusiastas no mercado dos CMSs, e também o repositório mais completo: milhares de *plugins*, *scripts* e *snippets*, abrangendo uma multiplicidade de funções, estão disponíveis *online* para *download* imediato e gratuito (Wordpress, 2015).

Para além dos *plugins* e das funções personalizadas, outra característica poderosa do Wordpress – embora não seja um conceito exclusivo deste CMS – é a robustez da funcionalidade dos *custom fields* (campos personalizados): “Os campos personalizados permitem que os utilizadores associem conteúdo personalizado e informações adicionais a cada publicação. O resultado é uma enorme flexibilidade em termos de organização do conteúdo, design e administração” (Coyier e Starr, 2012: 151).

Com a introdução dos campos personalizados na versão 1.2, os *developers* deixaram de ficar condicionados a uma só forma de apresentar o conteúdo. No entanto, esta funcionalidade ganha outra dimensão quando usada em conjunto com um *plugin* como o Advanced Custom Fields (Condon, 2015), que torna mais fácil, flexível e poderosa a utilização dos campos personalizados. Atualmente, a importância deste *plugin* é de tal ordem, que Ana Aires, uma das faces da WidgiLabs, uma empresa portuguesa especializada em Wordpress, assume que o Advanced Custom Fields integrado de raiz no sistema é uma das funcionalidades que mais gostava de ver implementadas no futuro (Freitas, 2014).

II.5 Segurança

Por razões óbvias, a segurança é um dos aspetos mais importantes a considerar durante o desenvolvimento e a manutenção de um produto Web. É cada vez mais

consensual de que não há *sites* 100 por cento seguros. Jeremiah Grossman (2008), um reputado especialista em segurança Web norte-americano, confirma esta ideia:

mesmo adicionando camadas sobre camadas de defesas, haverá sempre a hipótese de uma parte do sistema falhar. Além disso, colocar demasiada ênfase na obtenção da segurança “perfeita” vai resultar em retornos decrescentes. (...) O desafio é encontrar o equilíbrio certo entre os recursos investidos na segurança (tempo/dinheiro), de forma a reduzir adequadamente (não eliminar) a probabilidade de [um *site*] ser *hackeado*.

Este princípio aplica-se tanto ao Wordpress como a qualquer outra aplicação Web. Ainda assim, e ao contrário do que é muitas vezes sugerido, o Wordpress é uma plataforma bastante segura. Uma das características mais poderosas do Wordpress é a sua enorme comunidade de *developers* e utilizadores entusiastas. Qualquer nova vulnerabilidade descoberta é rapidamente resolvida por um dos milhares de *developers* especializados em Wordpress espalhados pelo mundo. Seja na forma de pequenos fragmentos de código (*snippets*), de atualizações de *software* ou de novos *plugins*, fica disponível em tempo recorde, para toda a comunidade e de forma gratuita, a solução para “aquela” última vulnerabilidade do sistema.

Mas a sua enorme popularidade é também a maior fragilidade, o grande calcanhar de Aquiles do Wordpress do ponto de vista da segurança: com o código à vista de todos, com tantos utilizadores e tantos *sites* assentes na plataforma, o Wordpress é naturalmente um dos alvos preferenciais dos *hackers*. Dito de uma maneira simples, os *hackers* “sabem como o sistema funciona e como atacá-lo” (Connelly, 2011: 151). Este facto deve servir de alerta para alguns aspetos que um Wordpress *developer* não pode negligenciar, porque se a segurança através da obscuridade é um método pouco aconselhável em qualquer sistema, muito menos o é num sistema *open source*. Compreender as principais vulnerabilidades e otimizar a segurança não é um trabalho difícil, mas requer um bom conhecimento da plataforma. Connelly (2011: 151) propõe uma lista com as 10 tarefas obrigatórias que devem ser levadas a cabo para aumentar a segurança de um *site* Wordpress:

1. Garantir que o acesso ao painel de administração não é violado
2. Fazer *backups* com frequência

3. Fazer *updates* astutamente
4. Neutralizar a conta <admin> que vem por defeito em muitas instalações
5. Corrigir as permissões dos ficheiros e das pastas
6. Esconder a versão do Wordpress
7. Alterar os prefixos das tabelas da base de dados
8. Configurar as chaves secretas
9. Negar o acesso ao ficheiro <config.php>
10. Restringir o acesso às pastas <wp-content> e <wp-includes>

Todas estas etapas são explicadas com minúcia em *Wordpress 3 Ultimate Security* (Connelly, 2011), um livro de 375 páginas completamente dedicado à temática da segurança em Wordpress. Se todos estes passos forem assegurados, a segurança da plataforma melhorará substancialmente. É claro que se o Wordpress estiver instalado num alojamento partilhado, uma parte da segurança dependerá do conhecimento, da perícia e das boas práticas dos prestadores do serviço de alojamento Web. Não vale de muito fechar a porta à chave deixando as janelas abertas.

Note-se que muitas destas tarefas podem ser executadas automaticamente por *plugins*, mas é desejável e recomendável que um *developer* compreenda bem as vulnerabilidades do sistema antes de utilizá-los. Não é aceitável que um *Web developer* assegure a segurança de um *site* apenas por instalar o plugin X ou Y; deve ser capaz de explicar claramente o que faz e por que razão o faz.

II.6 Usabilidade e acessibilidade

Independentemente da plataforma e das linguagens utilizadas para a criação e o desenvolvimento de um *site*, a adoção de boas práticas de usabilidade e acessibilidade deve ser uma preocupação permanente de um Web designer. O conceito de usabilidade Web está relacionado com a facilidade de uso e a qualidade da interação que um *site* proporciona aos seus utilizadores:

A usabilidade é um atributo de qualidade relacionado com o quão fácil é usar algo. Mais especificamente, refere-se à rapidez com que as pessoas podem aprender a usar alguma coisa, quão eficientes são ao usá-la, quão memorável é essa coisa, quão passível de erro é e quantos utilizadores gostam de usá-la. Se as pessoas não conseguem ou não vão usar um atributo, é preferível que ele não exista (Nielsen e Loranger, 2006: 365-371).

Tirando alguns casos excepcionais, os *sites* não são simples peças de decoração, objetos para serem apenas contemplados: têm como função distribuir informação e são produzidos em função de objetivos específicos. Estudos conduzidos por um dos nomes mais proeminentes da usabilidade Web, Jakob Nielsen, demonstram que os utilizadores preferem clareza a confusão, e que esta regra se aplica tanto ao conteúdo escrito como ao design (Nielsen e Loranger, 2006: 4045). Em última análise, de que serve um *site* se os seus utilizadores não conseguem manuseá-lo? A usabilidade Web não é mais do que facilitar o uso de um *site*, de modo a garantir que a sua utilização é eficaz, eficiente, satisfatória e memorável. Em suma, o objetivo último da usabilidade é assegurar que um *site* cumpre os objetivos a que se propõe.

O conceito de acessibilidade Web está relacionado com a capacidade de navegar na Internet e de aceder à informação independentemente das limitações físicas dos utilizadores. Garantir a acessibilidade significa que as pessoas com necessidades especiais podem fazer um uso pleno da Internet, ou seja, “conseguem perceber, entender, navegar e interagir com a Web” (Henry, 2006). Apesar das enormes potencialidades criadas pela Internet, há ainda uma série de barreiras⁹ que impedem uma navegação plenamente acessível, sobretudo porque os criadores de conteúdos Web não estão suficientemente sensibilizados para as questões da acessibilidade. O grande objetivo da acessibilidade Web é, pois, a remoção dessas barreiras, de modo a que “as pessoas com necessidades especiais possam também utilizar e contribuir para a Web” (Henry, 2006).

Foi nesse sentido que surgiu a necessidade de encontrar e promover uma série de princípios e diretrizes, que fossem consensualmente aceites, para ajudar todos aqueles que projetam, desenham, programam e atualizam os milhões de *sites* que hoje existem na Internet: “As Diretrizes de Acessibilidade para o Conteúdo da Web (WCAG) abrangem uma vasta gama de recomendações para tornar o conteúdo da Web mais acessível a um maior número de pessoas com incapacidades” (Umic, 2009). As WCAG estão assentes em quatro princípios-chave, que fornecem a infor-

⁹ Giorgio Brajnik (2011), um professor italiano especialista em acessibilidade Web, define barreira como “qualquer condição que atrapalha ou impede o progresso do utilizador na persecução de um objetivo, quando esse utilizador é portador de deficiências ou incapacidades”.

mação necessária para um utilizador aceder e usar os conteúdos da Web. Assim, o conteúdo Web tem de ser:

Perceptível – A informação e os componentes da interface de utilizador têm de ser apresentados aos utilizadores de formas perceptíveis. Isto significa que os utilizadores têm de ser capazes de compreender a informação apresentada (tem de estar visível a todos os seus sentidos); **Operável** – Os componentes da interface de utilizador e a navegação têm de ser operáveis. Isto significa que os utilizadores têm de ser capazes de funcionar com a interface (a interface não pode requerer uma interação que um utilizador não possa executar); **Compreensível** – A informação e a operação da interface de utilizador têm de ser compreensíveis. Isto significa que os utilizadores têm de ser capazes de compreender a informação e o modo de funcionamento da interface de utilizador (os conteúdos ou o funcionamento não podem ir para além da sua compreensão); **Robusto** – O conteúdo tem de ser suficientemente robusto para ser interpretado, com precisão, por uma grande variedade de agentes de utilizador, incluindo tecnologias de apoio. Isto significa que os utilizadores têm de ser capazes de aceder aos conteúdos à medida que as tecnologias avançam (à medida que as tecnologias e os agentes de utilizador evoluem, os conteúdos devem permanecer acessíveis) (Unidade Acesso, 2008b).

Se algum destes princípios não se confirmar, os utilizadores com incapacidades não conseguem fazer um uso pleno da Internet. Porém, embora sejam o alvo prioritário, as pessoas com necessidades especiais não são o único foco da acessibilidade Web. Ao proporcionar a redundância da navegação, em que o acesso ao conteúdo pode ser feito por mais do que uma forma, todas as pessoas acabam por beneficiar de uma maior flexibilidade de navegação proporcionada pelos *sites* mais acessíveis. É o caso dos idosos, que vão vendo as suas faculdades diminuírem com o tempo, ou das pessoas com necessidades especiais provisórias provocadas por acidentes ou intervenções cirúrgicas. A acessibilidade é, portanto, uma questão central nesta nova era digital em que vivemos. Para além de ser essencial para garantir a igualdade de acesso e de oportunidades, a acessibilidade Web diz respeito a todos, porque, mais tarde ou mais cedo, todos beneficiarão com ela.

CAPÍTULO III: QUESTÕES DE ORDEM METODOLÓGICA

Antes dos processos e das práticas diretamente relacionados com a execução do meu trabalho, tive, em primeiro lugar, de me familiarizar com os métodos de trabalho da instituição de acolhimento. Não se trata aqui de desresponsabilizar-me, ou de atribuir a responsabilidade a terceiros, mas de mencionar o óbvio: o estagiário está, compreensivelmente, obrigado a respeitar as regras de funcionamento internas da instituição de acolhimento.

Um dos métodos que mais estranhei, mas porventura um dos mais proveitosos, foram as reuniões diárias. Todos os dias, depois do almoço, havia uma reunião em que todos os colaboradores explicavam o trabalho que estavam a desenvolver e/ou que tinham desenvolvido nas últimas 24 horas. As reuniões vieram a revelar-se preciosas do ponto de vista pessoal e profissional. Em primeiro lugar, porque o facto de estarem todos os colaboradores em pé de igualdade ajudava a fomentar o espírito de entreajuda, contribuindo para que a minha integração fosse mais rápida. Em segundo lugar, porque me obrigavam a estruturar mentalmente o trabalho que estava a desenvolver, de modo a conseguir explicá-lo convenientemente. Especialmente o trabalho de programação envolve linguagens muito próprias, por vezes herméticas, que tornam a sua compreensão muito difícil. A obrigação de ter de explicar diariamente o que fazia a pessoas que não dominavam as linguagens Web, ou a linguagem mais específica do Wordpress, foi um excelente exercício de estruturação do conhecimento que, até então, se manifestava mais vezes de forma instintiva. Julgo que Albert Einstein terá dito que se não soubermos explicar algo de forma simples é porque não entendemos bem o que estamos a explicar. Apesar da dificuldade que senti em clarificar alguns tópicos mais específicos, considero que estas reuniões melhoraram substancialmente a minha capacidade de comunicá-los. Em terceiro lugar, porque, sendo as reuniões muitas vezes em inglês, eram momentos em que podia treinar mais a oralidade da língua inglesa. Por todas estas razões, sinto que as reuniões tiveram um valor inestimável.

Ainda antes do desenho, da composição dos elementos numa página ou da programação propriamente dita, uma parte importante do trabalho de um Web designer

ocorre nas fases anteriores da pesquisa, do estudo da audiência e da elaboração do diagrama conceptual e do design lógico. Porém, o trabalho de desenvolvimento Web que executei durante o estágio começava numa fase posterior. Quando o trabalho me chegava às mãos, já as fases de pesquisa tinham sido delineadas pela empresa. O meu trabalho consistia em transformar os protótipos, que me eram entregues em ficheiros XMind (Apêndice A e Apêndice B), em páginas Web assentes na plataforma Wordpress. Isto é, o meu trabalho foi mais de programação Web e menos de Web design. E aqui há duas diferenças a registar: o *site* da Azorean partiu de um tema Wordpress já existente, e o meu trabalho foi sobretudo de edição HTML, CSS e PHP, de refinamento e criação de funções e de limpeza de código; o *site* do Ziphius foi criado e desenvolvido de raiz e, por fim, adaptado para um tema Wordpress original. A adoção de um tema já comprado para o *site* corporativo não foi feita sem algumas reservas, mas acabou por se justificar pela urgência que a empresa manifestou: o *front end*¹⁰ tinha de estar concluído em cerca de duas semanas, altura em que a Azorean teria uma reunião importante com potenciais investidores.

Embora estes dois trabalhos tenham exigido abordagens diferentes, devido às especificidades de cada um, as duas experiências foram úteis para refletir sobre as diferentes fases do desenvolvimento de um produto Web baseado na plataforma Wordpress. As fases podem alterar-se em função do tipo de *site* que se está a desenvolver, mas há pelo menos cinco grandes etapas que o compõem: 1) conversão dos ficheiros HTML, PHP, CSS e Javascript para o formato Wordpress; 2) criação e configuração dos *custom post types* e das taxonomias; 3) instalação e/ou criação e configuração de *plugins*; 4) programação das opções do tema – para serem geridas a partir do painel de administração; e 5) configuração da segurança, SEO e otimização do desempenho. No desenvolvimento do *site* do Ziphius, a quarta etapa foi a mais demorada. Como o *site* ainda não tinha conteúdos textuais, fotográficos e videográficos, e não ficaria concluído durante o meu período de estágio, decidi criar diversas opções no tema para que os conteúdos pudessem ser inseridos e geridos mais tarde, de uma maneira simples e intuitiva, a partir do painel de administração. A ideia era garantir uma maior autonomia à empresa; deixá-la o menos dependente possível de

¹⁰ O *front end* de um *site* é composto por todos os elementos que o utilizador vê e com os quais interage diretamente.

um programador Web. Para isso, foi primeiro necessário pensar sobre quais seriam as opções e funções a disponibilizar no painel de administração, e de que forma seriam tornadas acessíveis. As minhas decisões tiveram em conta o conhecimento e o domínio que a pessoa responsável pela atualização dos conteúdos tinha da plataforma. Se, por um lado, mais opções disponíveis significam maior liberdade e flexibilidade, por outro, um menor número de opções tem a vantagem de facilitar a manipulação do sistema. Porque com a flexibilidade vem a complexidade: quanto mais opções uma interface tem, mais complexo e difícil é o seu manuseamento.

Outro trabalho que desenvolvi durante o estágio foi a programação da página “*Preview*” da App¹¹ Ziphius (Apêndice C). O método era simples: depois de recebidos os protótipos da página produzidos pelo designer da empresa, o meu trabalho consistia em convertê-los nas linguagens HTML e CSS, tendo presente as proporções relativas a três diferentes tamanhos de ecrã: grande, médio e pequeno. De forma a garantir a consistência do código, procurei, em primeiro lugar, perceber como estavam programadas as outras páginas da aplicação. Da mesma maneira que podemos utilizar várias estradas para chegarmos ao mesmo destino, também podemos utilizar diferentes abordagens de programação para chegarmos ao mesmo objetivo final. Raramente há apenas uma solução possível.

Ao analisar o código, chamou-me a atenção uma particularidade: todos os elementos estavam posicionados em modo absoluto. Não havia uma grelha ou um contentor que albergasse os elementos. Esta abordagem, sendo possível, é desaconselhável, porque dificulta futuras alterações. Se, por exemplo, algum dia for alterada ou adicionada uma resolução de um monitor, ou se houver a necessidade de modificar a posição relativa de alguns elementos, todos os elementos da página terão de ser alterados um a um. Se, pelo contrário, os elementos estiverem colocados em contentores posicionados em modo relativo, as futuras alterações podem ser confinadas apenas a esses contentores (Coyier, 2008). Esta é, de resto, a técnica utilizada por variadíssimas estruturas HTML/CSS, entre elas a do Bootstrap (2015). No entanto, como uma grande parte da aplicação já estava programada, optei por apenas

¹¹ Nome abreviado para “aplicação móvel”. É o *software* desenvolvido para ser instalado em dispositivos móveis, como telemóveis, *smartphones* ou *tablets*.

informar os meus responsáveis sobre este problema e introduzir os elementos também em modo absoluto, de forma a manter a programação da App consistente. Além do mais, uma alteração desta natureza, numa fase tão avançada do projeto, demoraria bastante tempo a concretizar.

As boas práticas da acessibilidade aconselham os Web designers/developers a considerá-la desde as primeiras fases do desenvolvimento de um produto Web. A razão é simples: quanto mais cedo se começa a configurar a acessibilidade de um *site*, menos tempo se perde na fase final. Por outro lado, não respeitando alguns aspetos fulcrais da acessibilidade desde o início, pode correr-se o risco de ser demasiado difícil e moroso corrigi-los numa fase posterior. De resto, um dos problemas mais comuns dos temas Wordpress *premium* é precisamente a fraca prestação em matéria de acessibilidade. Apesar da multiplicidade de funções e das características avançadas que oferecem, a acessibilidade é ainda um atributo descurado pela grande maioria deles. Por essa razão, a configuração e a otimização da acessibilidade do *site* da Azorean foram efetuadas *a posteriori*, numa altura em que o *site* já se encontrava *online* há algum tempo. De forma a analisar e a corrigir os problemas mais comuns, foi utilizado o validador automático português para as WCAG, AccessMonitor, desenvolvido pela Unidade Acesso da Fundação para a Ciência e a Tecnologia. A opção pelo AccessMonitor teve duas razões fundamentais: em primeiro lugar, porque este validador agrupa os diferentes problemas em categorias, tornando mais fácil a leitura dos dados; em segundo lugar, porque fornece uma escala numérica de avaliação, o que permite uma apreciação mais objetiva dos resultados. O AccessMonitor utiliza um índice numa escala de 1 a 10, representando o valor 10 uma adoção plena das boas práticas referidas pelas Diretrizes de Acessibilidade para o Conteúdo da Web (Unidade Acesso, 2008a). A análise apenas considerou os problemas mais comuns e recorrentes. Para uma análise mais detalhada, que examinasse contextos de utilização específicos, seria fundamental recorrer à validação manual e a testes por utilizadores com necessidades especiais. Ainda assim, a melhoria dos *rankings* da acessibilidade do *site* foi considerável: algumas páginas registaram melhoramentos na ordem dos 4 pontos (Apêndice D).

CAPÍTULO IV: CONCRETIZAÇÃO DO PLANO DE ATIVIDADES

Este capítulo descreve as principais atividades desenvolvidas durante o estágio. O desenvolvimento do *site* corporativo da YDreams, uma das atividades contempladas no plano inicial, acabou por não se concretizar, tendo sido substituído pela produção do *site* do Ziphius, o produto que a empresa se prepara para comercializar. De notar que as fases de criação, produção e desenvolvimento de um produto Web envolvem uma série de etapas e uma multiplicidade de disciplinas, pelo que é difícil conseguir sintetizar em poucas páginas todas as tarefas que de facto executei. Destacarei, por isso, aquelas que considero mais relevantes por área de atividade.

IV.1 Produção e desenvolvimento Web

O *Web development* foi a área de atividade que me ocupou mais tempo durante o estágio: produzi e desenvolvi dois *sites* assentes na plataforma Wordpress – o *site* corporativo da Azorean (Apêndice E) e o *site* do produto que a empresa está a desenvolver, Ziphius (Apêndice F) – e programei a página “*Preview*” da App Ziphius.

O *site* corporativo foi o primeiro trabalho que executei durante o estágio e o que mais tempo demorou a concretizar. O processo foi sobretudo composto pela programação de linguagens Web, designadamente HTML, CSS, PHP e Javascript (jQuery¹²). Foram várias as fases que constituíram o trabalho: hierarquização da tipografia; customização de diversas funções e elementos gráficos; criação de funcionalidades que o tema Wordpress original não tinha e que me iam sendo pedidas; eliminação de funções supérfluas; criação de páginas personalizadas (*custom post types*); SEO (*Search Engine Optimization*); e otimização da acessibilidade Web. Uma boa parte do estágio – cerca de 60 por cento da duração total – foi passado a trabalhar neste *site*, que foi desenvolvido a partir de um tema Wordpress já existente. É, então, natural que se pergunte como é que, trabalhando a partir de um tema já criado, se pode demorar tanto tempo. Não seria suposto que um tema *premium*

¹² “O jQuery (lê-se ‘djay-query’) é uma biblioteca Javascript (...) totalmente livre e de código aberto (...). De forma geral, tem como missão tornar o desenvolvimento Web mais rápido” (Soares, 2013: 9).

abreviasse o tempo de desenvolvimento de um sítio Web? A experiência que adquiri com este trabalho tornou claro que a adoção de um tema Wordpress já comprado pode não facilitar o desenvolvimento Web, mas, pelo contrário, dificultá-lo. Esta foi, porventura, uma das conclusões mais importantes que tirei desta experiência.

A programação da página “*Preview*” da aplicação móvel Ziphius acabou por ser um trabalho menor, uma vez que faltavam ainda uma série de elementos gráficos necessários para a conclusão do trabalho, e que fugiam da minha alçada. O processo consistiu em programar HTML e CSS da mesma forma que se programa uma página Web, utilizando um *software* – Unity (Coherent UI) – para converter as linguagens Web na linguagem de cada sistema operativo – Swift no caso do iOS, Java no do Android. Um método simples e eficaz, sobretudo para quem, como eu, não se sente muito confortável com as linguagens destes sistemas móveis.

O *site* do Ziphius foi o último trabalho que desenvolvi. Como me foi solicitado a cerca de duas semanas de concluir o estágio, e como ainda não tinham sido produzidos quaisquer conteúdos exceto as notícias, está ainda por terminar. Toda a programação foi pensada e executada deixando espaços reservados (*placeholders*) para a inserção desses conteúdos no futuro através do painel de administração. O processo de trabalho foi semelhante ao do *site* corporativo, com a exceção de que o tema Wordpress foi construído de raiz. De resto, quase todas as etapas que constituíram esse trabalho foram as mesmas, mas aqui não houve a necessidade de editar ou remover funções supérfluas, porque todas foram criadas de raiz em função dos objetivos do *site*.

IV.2 Problemas encontrados na customização de alguns elementos

O *site* corporativo da Azorean foi desenvolvido a partir do tema One (Theme Forest, 2014), um tema *premium* Wordpress que já tinha sido adquirido pela empresa. Esse facto condicionou bastante algumas etapas do desenvolvimento do *site*, uma vez que o tema original não contemplava algumas funções que me iam sendo pedidas. Nesse sentido, foi necessário introduzir código novo – sobretudo CSS, PHP e Javascript – e programar de raiz alguns componentes, mas também editar funções incluídas no

tema. Por exemplo, o tema One permitia colocar um vídeo do Vimeo ou do YouTube em modo *fullscreen* (a ocupar toda a janela do *browser*), de uma maneira simples e intuitiva, a partir do painel de administração. Porém, quando introduzi pela primeira vez o vídeo, deparei-me com dois problemas insolucionáveis: primeiro, a cor de fundo dos vídeos enquanto carregavam não podia ser alterada – tanto os vídeos do Vimeo como do YouTube têm o fundo preto por defeito e não dá para ser alterado; segundo, o posicionamento do vídeo em modo *fullscreen* estava condicionado aos vídeos alojados nesses *sites*. Como precisávamos de um fundo branco, não houve alternativa senão programar uma função que permitisse ao tema albergar vídeos alojados no próprio servidor, tirando partido do elemento `<video>` introduzido no HTML5.

Outro exemplo é a página “Ziphius” do mesmo *site*. De acordo com o protótipo que a empresa me entregou inicialmente, a ideia para esta página passava por apresentar uma fotografia do produto com fundo branco. Porém, alguns dias depois de o *site* ter ficado *online*, foi-me pedido que pensasse numa forma de os utilizadores poderem rodar a fotografia num ângulo de 360 graus. Não estando esta funcionalidade integrada no tema original, foi preciso pensar nas alternativas mais céleres que tinha ao meu dispor. Para além da criação e programação de um *custom post type*, foi necessário recorrer a uma livraria Javascript, especialmente concebida para rotações 3D, adaptá-la às necessidades do *site* e integrá-la no tema Wordpress. Com a ajuda de um colega estagiário, que produziu os *renders* 3D em poucos dias, tínhamos 100 fotografias, cada uma com um ângulo diferente, prontas para serem usadas na rotação.

Outras customizações passaram pela troca de fotografias à passagem do rato, na página “Team”, pela resolução de um problema de renderização tipográfica que ocorria no Chrome em versão Windows, pela configuração SEO e pela otimização da acessibilidade. Os elementos customizados tiveram ainda de ser reprogramados de modo a garantir-se a sua responsividade.

Todas estas customizações tiveram como consequência inevitável a adição de código HTML, CSS, PHP e Javascript ao já extenso código incluído no tema original, impondo uma pressão adicional ao desempenho geral do *site*. Foi por essa razão que, depois de ter dado por concluída toda a programação, passei à fase de limpeza de algum código supérfluo. Digo “algum”, porque simplesmente não é possível reti-

rar todo o código desnecessário de um tema que não conhecemos totalmente e num espaço de tempo tão curto. É um trabalho minucioso que pode demorar dias. E demorando demasiado tempo, deixa de fazer sentido utilizar um tema *premium*, porque uma das grandes vantagens de se trabalhar a partir de um tema que já existe é, teoricamente, a economia de tempo. Nessa medida, o meu objetivo enquanto *Web developer* passa por encontrar um compromisso entre o desempenho do *site* e as funções necessárias para o correto funcionamento do tema em função do tempo disponível para a conclusão do projeto.

IV.3 Outras atividades desenvolvidas

Uma das características mais estimulantes do desenvolvimento de um produto Web é a multiplicidade de disciplinas que estão envolvidas nesse trabalho. Por exemplo, a edição de imagem ou de vídeo é um trabalho que está normalmente associado à produção de *sites*. A este propósito, a Azorean pretendia um vídeo, que já tinha adquirido e editado, para ocupar todo o ecrã da página inicial do *site* corporativo da empresa. Contudo, o vídeo era demasiado grande e pesado, o que comprometia substancialmente o desempenho do *site*. Por essa razão, fui forçado a editá-lo, recorrendo aos programas de edição de vídeo da Adobe, Premiere e After Effects, de modo a torná-lo mais pequeno e mais leve. O objetivo da edição era encurtar o vídeo e, ao mesmo tempo, encontrar o ponto de *loop* perfeito, para que as repetições pudessem ser o mais naturais possível.

Como já expliquei no Capítulo III deste relatório, o trabalho de desenvolvimento Web que executei foi mais de programação e menos de Web design. Ainda assim, trabalhos como o desenho dos cabeçalhos da *newsletter* institucional da Azorean, a configuração da acessibilidade do *site* corporativo e o desenho de algumas secções do *site* do Ziphius, especialmente a *home page* e a página “News”, foram alguns dos trabalhos que executei e que estão mais relacionados com o Web design e a experiência do utilizador – este último mais no âmbito da acessibilidade.

Outra atividade que desenvolvi durante o estágio foi a locução, gravação e edição de áudio. Esta área pouco ou nada está relacionada com o curso de mestrado

em Novos Media e Práticas Web, mas é uma área a que estou ligado profissionalmente há muitos anos. O trabalho de locução foi realizado no âmbito de um concurso de *gadgets* ao qual a Azorean concorreu com um produto chamado “Automarine”. Os trabalhos de gravação, edição áudio e sonoplastia foram executados no âmbito do projeto MOnarCH (Monarch, 2013).

CONCLUSÃO

O estágio foi um período em que, para além de outros trabalhos de menor dimensão, pude desenvolver dois *sites* assentes em plataforma Wordpress. Durante esse tempo, tive a oportunidade de refletir sobre alguns benefícios e desafios do sistema Wordpress no desenvolvimento de sítios Web.

Continua a ser difícil dar uma resposta categórica de quais são as vantagens do Wordpress em relação a outros sistemas de gestão de conteúdos. Quando comparado a sistemas pagos, a resposta não pode deixar de ser a gratuidade e a extensibilidade. Sobretudo quando utilizado em *sites* de conteúdo predominantemente editorial, como jornais ou revistas *online*, o desempenho do Wordpress é irrepreensível e semelhante a muitos outros sistemas com licenças de utilização bastante caras. Não é por acaso que *sites* como o *The New Yorker*, *BBC America*, *Sony Music* ou *Fortune* são assentes nesta plataforma. Mas o Wordpress tem uma vantagem suplementar: a extensibilidade. Expandir o sistema pode ser um processo tão simples como instalar alguns *plugins*, muitos deles também gratuitos.

Contudo, a resposta torna-se mais difícil quando comparamos o Wordpress com outros dois CMSs populares e igualmente gratuitos, o Joomla e o Drupal. Neste caso, a maior popularidade do Wordpress, conquistada quando ainda era um sistema de *blogging*, é a sua principal arma. Por um lado, com tantos utilizadores já familiarizados com a plataforma, o Wordpress proporciona uma curva de aprendizagem mais rápida. Por outro, a maior comunidade *online* de *developers* e entusiastas no mercado dos CMSs garante uma extensibilidade superior à da concorrência: milhares de *plugins*, *scripts* e *snippets*, abrangendo uma multiplicidade de funções, estão disponíveis *online* para *download* imediato e gratuito.

Mas a enorme popularidade é também a sua maior fragilidade, o grande calcanhar de Aquiles do Wordpress do ponto de vista da segurança: com o código à vista de todos, com tantos utilizadores e *sites* assentes na plataforma, o Wordpress é naturalmente um dos alvos preferenciais dos *hackers*. Não é que o sistema seja intrinsecamente mais vulnerável do que outros; é apenas mais apetecível por ser

utilizado por mais pessoas. Esta característica tem um aspeto positivo: força os *Web developers* a configurar convenientemente a segurança da plataforma, porque não é possível adotar apenas métodos de segurança através da obscuridade – baseados na falsa suposição de que ninguém poderá descobrir e enganar o sistema.

Embora os dois trabalhos tenham exigido abordagens diferentes, porque um partiu de um tema Wordpress que já existia e outro foi desenvolvido de raiz, as duas experiências foram também úteis para refletir sobre as diferentes fases do desenvolvimento de um produto Web baseado na plataforma Wordpress. As fases podem alterar-se em função do tipo de *site* que se está a desenvolver, mas há pelo menos cinco grandes etapas que o compõem: 1) conversão do HTML/PHP/CSS/Javascript para um tema Wordpress; 2) criação e configuração dos *custom post types* e das taxonomias; 3) instalação e/ou criação e configuração de *plugins*; 4) programação das opções do tema – para serem geridas a partir do painel de administração; e 5) configuração da segurança, SEO e otimização do desempenho.

Ao longo do desenvolvimento destes dois *sites* em Wordpress, ocorreu-me invariavelmente uma pergunta: “será preferível criar um tema de raiz, ou, pelo contrário, trabalhar a partir de um tema já existente?” A experiência que adquiri durante o estágio leva-me a concluir que as duas hipóteses são viáveis. Trabalhar de uma maneira ou de outra depende dos objetivos do projeto, do tempo disponível e do grau de customização pretendido. Se a urgência na conclusão do trabalho é um fator determinante, se o tema a partir do qual vamos trabalhar tem uma hierarquia de informação semelhante à que pretendemos, se as funções do tema são as que necessitamos, e se a acessibilidade for encarada como uma questão secundária, a opção por um tema já criado pode ser uma solução perfeitamente viável. Se, por outro lado, o *site* exigir um grau de customização elevado, a criação de funcionalidades específicas e altos padrões de acessibilidade, a opção pelo desenvolvimento de um tema de raiz é a melhor solução.

O exemplo do *site* corporativo é paradigmático e ajuda a ilustrar melhor esta ideia. Como já referi atrás, o *site* da Azorean foi desenvolvido a partir de um tema *premium* que já tinha sido adquirido pela empresa, porque havia urgência na conclusão do trabalho. O problema é que, durante o período de desenvolvimento, foram

sendo pedidas novas funcionalidades que o tema original não tinha. Ora, novas funcionalidades significam, por um lado, mais tempo de desenvolvimento e, por outro, mais código adicionado ao já extenso arsenal de código incluído num tema *premium* Wordpress, impondo uma pressão suplementar ao desempenho geral do *site*. O tempo que se economizou numa fase inicial da produção acabou por ser ultrapassado, numa fase posterior, pela adição de novas funções e otimização do desempenho e da acessibilidade. No fim, o *site* tinha mais código e tinha demorado mais tempo a ser desenvolvido. Esta abordagem teve apenas um benefício: a partir da segunda semana, o *layout* do *site* tinha já uma aparência bastante agradável, com uma grande parte da responsividade configurada. O problema é que esta vantagem acarretou demasiados custos do ponto de vista do desempenho.

No futuro, de forma a assegurar um desempenho mais eficiente, seria importante valorizar mais a fase do planeamento, e, se possível, integrar o *Web developer* nessa etapa inicial e tão decisiva. Ninguém está mais habilitado do que ele para decidir sobre quais são as melhores estratégias de desenvolvimento Web a adotar. Quanto mais estruturada e detalhada for a fase de planificação, menos tempo demorará a ser concluído e melhor será o desempenho do produto final.

Estas foram algumas das reflexões que as dinâmicas práticas do estágio me proporcionaram. Sinto-me hoje mais competente e confiante para executar trabalhos de programação e desenvolvimento Web. Este foi, sem dúvida, o maior benefício que o período de estágio me ofereceu. Além disso, foram três meses em que pude dedicar-me quase exclusivamente ao desenvolvimento de *sites* em plataforma Wordpress, um tempo que contribuiu para melhorar as minhas práticas e aprender mais, porque a aprendizagem é um processo contínuo que nunca se dá por concluído.

BIBLIOGRAFIA

- Arnheim, R. (1974). *Art and Visual Perception: A Psychology of the Creative Eye*. Berkeley: University of California Press.
- Beaird, J. (2010). *The Principles of Beautiful Web Design*. SitePoint Pty Ltd.
- Chin, P. (2008). *Best Practices for Developing a Web Site – An internet.com Project Management eBook*. Internet.com.
- Connelly, O. (2011). *Wordpress 3 Ultimate Security*. Birmingham: Packt Publishing.
- Converse, T.; Park, J.; Morgan, C. (2005). *PHP5 and MySQL Bible*. Indianapolis: Wiley Publishing, Inc.
- Coyier, C.; Starr, J. (2012). *Digging Into Wordpress*. digWP.com.
- Dabner, D. (2006). *Guia das Artes Gráficas: Design e Layout*. Barcelona: Gustavo Gili.
- Damas, L. (2005). *SQL*. Lisboa: FCA.
- Danesh, A. (2004). *JavaScript in 10 Steps or Less*. Indianapolis: Wiley Publishing, Inc.
- Figueiredo, B. (2004). *Web Design*. Lisboa: FCA.
- Friedman, J. (2010). *Web Designer's Guide to Wordpress*. Berkeley: New Riders.
- Garrand, T. (2006). *Writing for Multimedia and the Web*. Oxford: Focal Press.
- Gasston, P. (2013). *The Modern Web*. San Francisco: No Starch Press.
- Genuis, S. K. (2004). Web Site Usability Testing: A Critical Tool for Libraries Usability testing in library. *Feliciter*, (4), 161–164.
- Hedengren, T. D. (2010). *Smashing Wordpress - Beyond the Blog*. West Sussex: John Wiley & Sons, Ltd.
- Harper, S.; Yesilada, Y. (Eds.) (2008). *Web Accessibility - A Foundation for Research*. London: Springer.
- Hogan, B. P.; Warren, C.; Weber, M.; Johnson, C.; Godin, A. (2012). *Web Development Recipes*. Dallas: The Pragmatic Programmers, LLC.
- Jardí, E. (2014). *Pensar com imagens*. São Paulo: Gustavo Gili.
- jQuery Community Experts (2010). *jQuery Cookbook*. Sebastopol: O'Reilly Media.

- Kane, J. (2012). *Manual dos Tipos*. Barcelona: Gustavo Gili.
- Kress, G.; Leeuwen, T. (2006). *Reading Images: The Grammar of Visual Design*. Oxon: Routledge.
- Krug, S. (2010). *Rocket Surgery Made Easy - The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. Berkeley: New Riders Press.
- Lawrence, D.; Tavakol, S. (2007). *Balanced Website Design - Optimising Aesthetics, Usability and Purpose*. London: Springer.
- Lupton, Ellen; Phillips, J. C. (2009). *Novos Fundamentos do Design*. São Paulo: Cosac Naify.
- Lupton, E. (2004). *Thinking With Type*. New York: Princeton Architectural Press.
- Macdonald, M. (2013). *WordPress - The Missing Manual*. Sebastopol: O'Reilly Media.
- Maeda, J. (2006). *The Laws of Simplicity*. The MIT Press.
- Morville, P.; Rosenfeld, L. (2006). *Information Architecture for the World Wide Web*. Sebastopol: O'Reilly Media.
- Muller-Brockmann, J. (2014). *Sistemas de grelhas*. Barcelona: Gustavo Gili.
- Munari, B. (2009). *Design e Comunicação Visual*. Lisboa: Edições 70.
- Nielsen, J.; Loranger, H. (2006). *Prioritizing Web Usability*. [livro eletrónico] Berkeley: New Riders.
- Pereira, A.; Poupa, C. (2011). *Linguagens Web*. Lisboa: Sílabo.
- Remoaldo, P. (2011). *CSS3*. Lisboa: FCA.
- Rubin, J.; Chisnell, D. (2008). *Handbook of Usability Testing*. Indianapolis: Wiley Publishing, Inc.
- Serrão, C.; Marques, J. (2009). *Programação com PHP 5.3*. Lisboa: FCA.
- Sitecore (2013). *How to Choose the Best Web Content Management System for Customer Experience Management: A Guide for Both Marketers and Developers*. Sitecore.
- Soares, L. (2013). *jQuery - A sua Biblioteca Javascript*. Lisboa: FCA.
- Stocks, E. J. (2009). *Sexy Web Design*. Victoria: SitePoint Pty Ltd.
- Tavares, F. (2012). *Desenvolvimento de Aplicações em PHP*. Lisboa: FCA.

- Thatcher, J.; Burks, M.; Kirkpatrick, A.; Lawson, B. (2006). *Web Accessibility - Web Standards and Regulatory Compliance*. New York: Friends of.
- Ullman, L. (2008). *PHP 6 and MySQL 5 for Dynamic Web Sites*. Berkeley: Peachpit Press.
- Wallace, H. R. (2010). *WordPress 3 Site Blueprints*. Birmingham: Packt Publishing.
- Williams, B.; Damstra, D.; Stern, H. (2013). *Professional Wordpress Design and Development*. Indianapolis: John Wiley & Sons, Inc.
- Williams, B.; Richard, O.; Tadlock, J. (2011). *Professional WordPress Plugin Development*. Indianapolis: Wiley Publishing, Inc.
- Williams, R. (2004). *The Non-Designers Design Book*. Berkeley: Peachpit Press.

WEBGRAFIA

- Azorean – Aquatic Technologies (2013). Kickstarter >> Ziphius: The Aquatic Drone. *Kickstarter* [Em linha]. Disponível em: <https://www.kickstarter.com/projects/ziphius/ziphius-the-aquatic-drone>, [Consult. 28 novembro 2014].
- Bootstrap (2015). Bootstrap · The world’s most popular mobile-first and responsive front-end framework. *Bootstrap* [Em linha]. Disponível em: <http://getbootstrap.com>, [Consult. 12 fevereiro 2015].
- Brajnik, G. (2011). Barrier Walkthrough. *University of Udine* [Em linha]. Disponível em: <http://sole.dimi.uniud.it/~giorgio.brajnik/projects/bw/bw.html>, [Consult. 5 novembro 2013].
- Condon, E. (2015). ACF { Advanced Custom Fields Plugin for WordPress. *Advanced Custom Fields* [Em linha]. Disponível em: <http://www.advancedcustomfields.com>, [Consult. 6 fevereiro 2015].
- Coyier, C. (2008). Absolute Positioning Inside Relative Positioning. *CSS-Tricks* [Em linha]. Disponível em: <http://css-tricks.com/absolute-positioning-inside-relative-positioning>, [Consult. 9 fevereiro 2015].
- Freitas, J. (2014). Entrevista a Ana Aires: “Gostava que o plugin Advanced Custom Fields estivesse integrado no WordPress”. *Wordpress Portugal* [Em linha]. Disponível em: <http://wp-portugal.com/2014/12/10/entrevista-ana-aires-gostava-que-o-plugin-advanced-custom-fields-estivesse-integrado-no-wordpress>, [Consult. 12 dezembro 2014].
- Gavaldà, M. (2014). The History of WordPress, its Ecosystem and Community. *Kinsta* [Em linha]. Disponível em: <https://kinsta.com/learn/wordpress-history>, [Consult. 6 fevereiro 2015].
- Grossman, J. (2008). Jeremiah Grossman: 100% secure websites. *Jeremiah Grossman Blogspot* [Em linha]. Disponível em: <http://jeremiahgrossman.blogspot.pt/2008/03/100-secure-websites.html>, [Consult. 6 fevereiro 2015].
- Henry, S. L. (2006). Understanding Web Accessibility. *uiAccess* [Em linha]. Disponível em: <http://uiaccess.com/understanding.html>, [Consult. 8 novembro 2013].
- Mening, R. (2013). CMS comparison - WordPress vs Joomla vs Drupal. *Website Setup* [Em linha]. Disponível em: <http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal>, [Consult. 9 março 2015].

- Monarch (2013). MOnarCH • Multi-Robot Cognitive Systems Operating in Hospitals. *Monarch* [Em linha]. Disponível em: <http://monarch-fp7.eu/pt>, [Consult. 18 novembro 2014].
- Powell, B. (s.d.). Pros and Cons of Joomla, WordPress, and Drupal. *About Tech* [Em linha]. Disponível em: <http://cms.about.com/od/cms-basics/a/Pros-And-Cons-Of-Joomla-Wordpress-And-Drupal.htm>, [Consult. 9 março 2015].
- Rackspace Support (2013). CMS Comparison: Drupal, Joomla and Wordpress | Knowledge Center. *Rackspace* [Em linha]. Disponível em: http://www.rackspace.com/knowledge_center/article/cms-comparison-drupal-joomla-and-wordpress, [Consult. 9 março 2015].
- Santos, A. (2013). Insert Coin: New Challengers winner Ziphius backstage at Expand (video). *Engadget* [Em linha]. Disponível em: <http://www.engadget.com/2013/03/18/insert-coin-new-challengers-winner-ziphius-backstage-expand-video>, [Consult. 28 novembro 2014].
- Storm, D. (2012). Hack Yourself First: Interview Jeremiah Grossman. *Computer World* [Em linha]. Disponível em: <http://www.computerworld.com/article/2472290/security0/hack-yourself-first-interview-jeremiah-grossman.html>, [Consult. 6 fevereiro 2015].
- Theme Forest (2014). WordPress - One - The Creative Multipurpose Portfolio theme. *Theme Forest* [Em linha]. Disponível em: <http://themeforest.net/item/one-the-creative-multipurpose-portfolio-theme/7624003>, [Consult. 28 novembro 2014].
- Umic. (2009). Diretrizes de Acessibilidade para o Conteúdo da Web 2.0 (WCAG 2.0 do W3C) Publicadas em Português. *UMIC – Agência para a Sociedade do Conhecimento* [Em linha]. Disponível em: http://www.unic.pt/index.php?option=com_content&task=view&id=3217&Itemid=38, [Consult. 8 novembro 2013].
- Unidade Acesso (2008a). AccessMonitor. *Unidade Acesso* [Em linha]. Disponível em: <http://www.acessibilidade.gov.pt/accessmonitor>, [Consult. 30 outubro 2014].
- Unidade Acesso (2008b). Diretrizes de Acessibilidade para o Conteúdo da Web (WCAG) 2.0. *Unidade Acesso* [Em linha]. Disponível em: <http://www.acessibilidade.gov.pt/w3/TR/WCAG20>, [Consult. 6 maio 2014].
- W3Techs (2015). Market share trends for content management systems, February 2015. *Web Technology Surveys* [Em linha]. Disponível em: http://w3techs.com/technologies/history_overview/content_management, [Consult. 4 fevereiro 2015].
- Wordpress (2015). Plugin Directory. *Wordpress.org* [Em linha]. Disponível em: <https://wordpress.org/plugins>, [Consult. 7 fevereiro 2015].

Wordpress Codex (2015a). Function Reference. *Wordpress.org* [Em linha]. Disponível em: http://codex.wordpress.org/Function_Reference, [Consult. 7 fevereiro 2015].

Wordpress Codex (2015b). Template Hierarchy. *Wordpress.org* [Em linha]. Disponível em: http://codex.wordpress.org/Template_Hierarchy, [Consult. 7 fevereiro 2015].

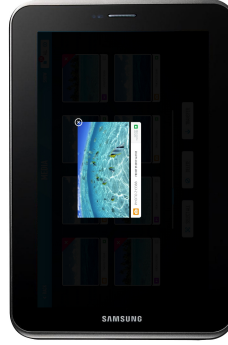
APÊNDICE C: PÁGINA “PREVIEW” DA APP ZIPHIUS

ZIPHIUS APP “MEDIA” PROPORTIONS TABLE

SMARTPHONE (DEFAULT)



TABLET MINI (DEFAULT)



TABLET (DEFAULT)



APÊNDICE D: RELATÓRIO DE ACESSIBILIDADE “AZOREAN.EU”

Azorean.eu - Avaliação automática da acessibilidade

Estes são os resultados obtidos depois de otimizadas todas as páginas. No caso da Home Page, a acessibilidade foi melhorada em quase 4 pontos. Noutras não foi possível este nível de otimização, porque o código é gerado dinamicamente pelo tema Wordpress ou pelos plugins utilizados. De todo o modo, os resultados são francamente positivos. A título de exemplo, o jornal online português com melhor índice de acessibilidade é o Expresso, que, ainda assim, apresenta um índice negativo - 4,4 numa escala de 1 a 10.

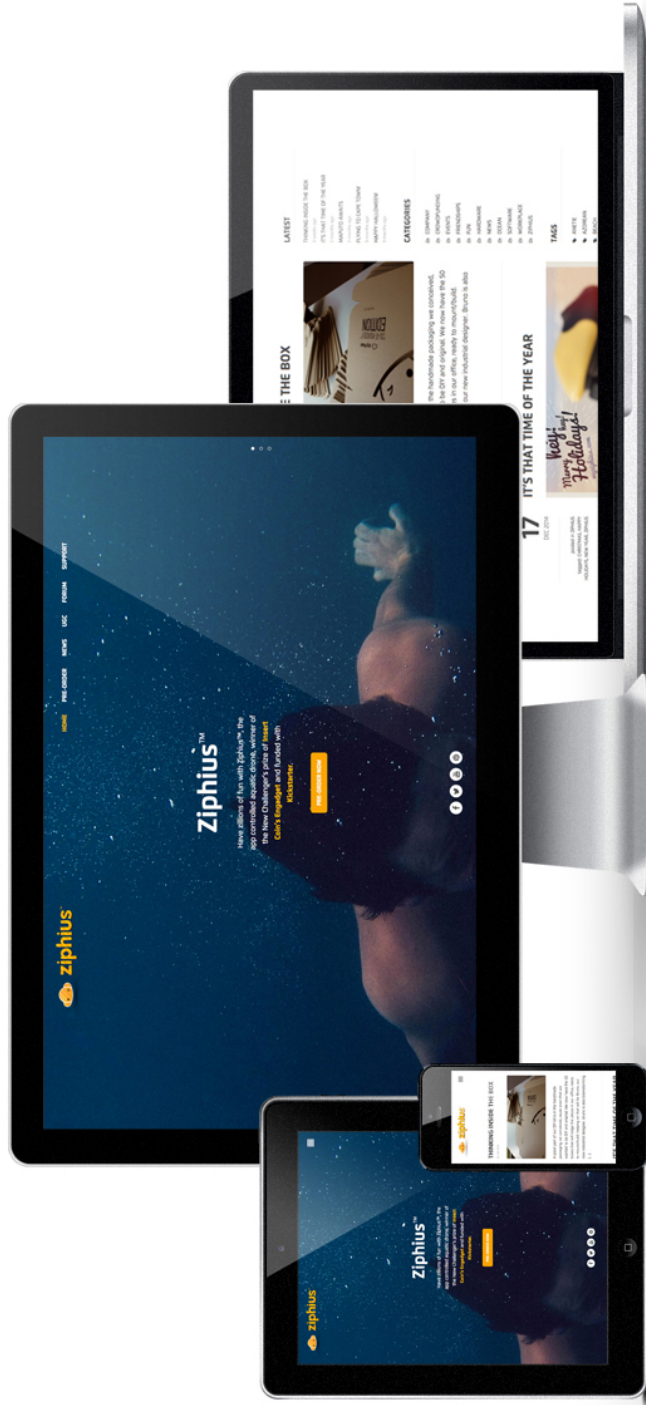
Páginas	Índice AccessMonitor *	Notas
http://azorean.eu	9	Acessibilidade melhorada em cerca de 4 pontos: A página original tinha 5,5 pontos.
http://azorean.eu/ziphius	7,9	Os atributos ID repetidos são uma consequência da programação dos criadores do template.
http://azorean.eu/ziphius/get-yours/	8	O atributo ID repetido é uma consequência da programação dos criadores do template.
http://azorean.eu/ziphius/get-yours/reserve/	7,6	O plugin utilizado para criar e gerir formulários não inclui a etiqueta <label> e o atributo <i>title</i>
http://azorean.eu/ziphius/get-yours/pre-order/	9	
http://azorean.eu/investor-relations/	6,9	Os atributos ID repetidos são uma consequência da programação dos criadores do template.
http://azorean.eu/news/	6,7	O atributo <i>alt</i> nulo e os links "Read More" são uma má prática de acessibilidade. Os criadores do template não tiveram isso em conta.
http://azorean.eu/team/	9	
http://azorean.eu/contacts/	6,8	Os atributos ID repetidos e a incorreta marcação de formulários são uma consequência da programação dos criadores do template.
Resultado global	7,88	

* Escala de 1 a 10. Disponível em <http://www.acessibilidade.gov.pt/accessmonitor/>

APÊNDICE E: ARTE FINAL “AZOREAN.EU”



APÊNDICE F: ARTE FINAL “MYZIPHIOUS.COM”



APÊNDICE G: FICHA DA EMPRESA

Designação Social	AZOREAN, Aquatic Technologies, SA
Presidente da Administração/CEO:	Edmundo Nobre
Endereço:	Rua da Pranchinha, nº 92
Localidade:	Ponta Delgada
Código Postal:	9500-331 Ponta Delgada
Telefone:	+351 210 314 000
Fax:	+351 210 314 001
E-mail:	hello@yatlantic.com
N.º de Identificação de Pessoa Coletiva:	509 951 490
CAE Principal:	26200-R3
N.º de colaboradores:	5
Volume de Negócios:	A Azorean está na fase de desenvolvimento de produto pelo que este valor não é demonstrativo
Principais clientes:	A Azorean está na fase de desenvolvimento de produto pelo que este valor não é demonstrativo
Principais projetos:	Desenvolvimento do Ziphius, um drone aquático de superfície
Parcerias internacionais:	Contactos internacionais para venda e distribuição do produto, assim como com universidades e laboratórios de ID no âmbito de projectos europeus
Website:	http://azorean.eu