



Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Patrícia Alexandra Gameiro Lopes

POR UM PIXEL

Relatório de Estágio

Orientado por:

Doutor João Patrício – Instituto Politécnico de Tomar
Miguel Abreu – Sketchpixel

Relatório de Estágio apresentado ao Instituto Politécnico de Tomar
para cumprimento dos requisitos necessários à obtenção do
grau de Mestre em Produção de Conteúdos Digitais

À minha família: Mãe e Irmão

RESUMO

O presente relatório descreve e analisa o trabalho realizado durante o segundo ano letivo do Mestrado em Produção de Conteúdos Digitais. Neste último ano foi realizado um estágio na Sketchpixel, empresa cujo objetivo é ascender no mercado das novas tecnologias, indo ao encontro das necessidades dos clientes e consumidores. A análise aqui exposta visa traduzir textualmente os desafios, pensamentos e convicções no exercício desta primeira experiência profissional na área dos Conteúdos Digitais.

São apresentados neste relatório os projetos desenvolvidos ao longo do estágio na empresa. Todos eles têm uma natureza muito distinta que vão desde a programação, à edição de imagem, passando pela criação de vídeo. Esta diversidade de trabalhos reflete também um pouco do plano de estudos do Mestrado, cujo principal foco é nos Audiovisuais e Produção dos Média.

Pretende-se demonstrar ainda a necessidade de equipas multidisciplinares na produção de conteúdos digitais completos, visto que exigem muitas vezes trabalho em diferentes áreas. Só assim é possível responder às exigências do público.

Palavras-chave: HTML5; CSS3; JavaScript; Motion Graphics; Imagem; Multidisciplinaridade.

ABSTRACT

This report describes and analyzes the work done during the second academic year of the Masters in Digital Media Production. During this past year, an internship at Sketchpixel took place. Sketchpixel is a company with headquarters in Braga that aims at rising the market for new technologies, meeting the needs of customers and consumers. This report documents the challenges and beliefs behind this first professional experience in the area of Digital Content.

In this report the projects developed during the internship in the company are presented. The projects have a very different nature, going from programming to image editing and creation of video. This diversity also reflects the Master syllabus, whose main focus is on Audiovisual and Media Production.

The report also aims to prove the need for multidisciplinary teams as far as production of full digital content is concerned, as quite often work in different areas is required. This is a critical condition for the fulfilment of the consumer demands.

Keywords: HTML5 ; CSS3 ; JavaScript ; Motion Graphics ; image; Multidisciplinary.

AGRADECIMENTOS

Este relatório representa não só a obtenção de mais um nível acadêmico mas é também o resultado de um enorme esforço e dedicação e da colaboração das mais diversas pessoas, às quais pretendo aqui manifestar a minha gratidão pelo apoio e a ajuda.

Em primeiro lugar, ao Doutor João Patrício tanto pela orientação neste segundo ano, o acompanhamento e organização na elaboração do presente relatório, como também pelo conhecimento e confiança transmitidos no primeiro ano de mestrado.

Em seguida, agradeço à *Sketchpixel* e a toda a sua equipa pela hospitalidade e simpatia no mês que passei em Braga. Um agradecimento sincero ao Miguel Abreu pela possibilidade da realização do estágio e pelo um grande voto de confiança no meu trabalho.

Agradeço aos meus colegas, que se tornaram grandes amigos, do Laboratório do Q207: Andreia Remédios, Pedro Isidoro e André dos Santos. Quero deixar aqui um grande reconhecimento ao Paulo Chorinca, *team leader*, por todos os conselhos e conhecimentos transmitidos ao longo do estágio, mas principalmente na parte letiva deste mestrado.

Aos meus amigos Míticos e companheiras Nádia Francisco e Carina Rocha que estiveram sempre presentes: nas horas de estudo e nos momentos de diversão.

Por fim, um agradecimento especial à minha Mãe, Irmão e Cunhada por todo o envolvimento, esforço e dedicação em tudo na minha vida. Obrigada por estarem sempre presentes!

Índice

1	Introdução	1
2	<i>Sketchpixel</i>	3
2.1	Relação com a empresa	3
2.2	Contextualização e motivação	3
3	Estado da arte	5
3.1	HTML5	5
3.2	CSS3	10
3.3	JavaScript	16
3.3.1	<i>Frameworks</i>	17
3.4	<i>Motion Graphics</i>	18
3.5	<i>Bitmap vs. Vector</i>	19
4	XPhoto	23
4.1	Tecnologias Utilizadas	23
4.2	Processo de Desenvolvimento	23
4.3	Considerações Finais	28
5	Aweshome	31
5.1	Tecnologias Utilizadas	31
5.2	Processo de Desenvolvimento	31
5.3	Considerações Finais	38
6	Motion Graphics	39

6.1	Tecnologias Utilizadas	39
6.2	Processo de Desenvolvimento	39
6.3	Considerações Finais	42
7	<i>WORLD OF DISCOVERIES - 'CEUTA ONTEM. CEUTA HOJE.'</i>	45
7.1	Livro Digital: <i>Crónica da Tomada de Ceuta</i>	45
7.2	Vinil para Pirâmide Holográfica	50
7.3	<i>Layouts: 'Ceuta em 1415' 'Ceuta Portuguesa'</i>	52
7.4	Considerações Finais	54
8	Conclusão	57
	Referências	59

Lista de Figuras

1	<i>Tags</i> usadas anteriormente com recurso a <i>div</i>	8
2	Novos elementos semânticos do HTML5	8
3	Lista de pesos existente para a fonte <i>Open Sans</i> (fonte utilizada no projeto) .	24
4	Instruções para a utilização de fontes	25
5	Utilização da propriedade <i>border-radius</i> na <i>tag div</i>	26
6	Exemplo da utilização de uma caixa com imagem de fundo e com a sua repetição e posição definidas	27
7	Utilização de uma <i>label</i> associada a um <i>input</i> para adquirir diferentes estados	28
8	Primeiros <i>layouts</i> do projeto concebidos	29
9	Reformulação do projeto ao nível da estrutura e do <i>layout</i>	30
10	Primeira versão da <i>homepage</i> do <i>Aweshome</i>	32
11	Página de elementos individuais da plataforma <i>Aweshome</i>	33
12	Caixas com a aplicação das propriedades de animação e transição através do pseudo-seletor <i>:hover</i>	34
13	Formulário para a introdução de um anúncio na área interna da plataforma .	36
14	Detalhes de uma propriedade anunciada no <i>Aweshome</i> , visível apenas para utilizadores registados	37
15	<i>Storyboard</i> para o vídeo <i>motion graphics</i>	41
16	Ecrãs para definição da imagem gráfica	42
17	Exemplo exemplificativo da animação em <i>keyframes</i> no <i>Adobe After Effects</i>	42
18	<i>Layout</i> de apresentação dos excertos traduzidos	47

19	Pena animada junto ao excerto a ser traduzido (lado direito)	48
20	<i>Layout</i> inicial de apresentação do <i>touchscreen</i>	49
21	Imagens finais para impressão em vinil	51
22	<i>Layout</i> ‘Ceuta em 1415’	54
23	<i>Layout</i> ‘Ceuta Portuguesa’	54
24	Sala da exposição ‘CEUTA ONTEM. CEUTA HOJE.’ no museu <i>World of Discoveries</i> no Porto Fonte: Página da <i>Sketchpixel</i> no <i>Facebook</i> ¹	55

Lista de Abreviaturas e Siglas

CSS *Cascading Style Sheets*

GIF *Graphics Interchange Format*

GPS *Global Positioning System*

HTML *Hypertext Markup Language*

PHP Originalmente, *Personal Home Page* e utilizado atualmente como um acrónimo recursivo para *Hypertext Preprocessor*

PNG *Portable Network Graphics*

SEO *Search Engine Optimization*

WHATWG *Web Hypertext Application Technology Working Group*

W3C *World Wide Web*

XHTML *eXtensible HyperText Markup Language*

XML *eXtensible Markup Language*

1 | Introdução

O gosto pela comunicação, partilha de informação e o desejo de conhecimento transferiram praticamente todos os conteúdos físicos para o mundo digital. A informação existe em qualquer lugar de forma a ser divulgada, partilhada, recolhida e solicitada. A chegada dos dispositivos móveis afirmou este cenário, incluindo mais um ecrã no quotidiano.

Este excesso de novos aparelhos que surgem diariamente permitem ao público desempenhar um papel ativo na partilha de informação e conhecimento. Foi ainda convertido o gosto do físico, duradouro e definitivo pelo tempo real, pelo agora, pelo efémero.

This remarkable ability to absorb and interpret visual information is the basis of industrial society and is becoming even more important in the information age. It is not a natural human attribute but a relatively new learned skill. [...] The hyper-stimulus of modern visual culture from the nineteenth century to the present has been dedicated to trying to saturate the visual field, a process that continually fails as we learn to see and connect faster and faster (Mirzoeff, 1999, 5).

É neste contexto, e com esta alteração de paradigma da sociedade, que surge o produtor de conteúdos digitais. É necessário existir uma resposta rápida por parte da produção de modo a que a informação seja adequada às novas plataformas e acessível em qualquer lugar. As audiências são cada vez mais seletivas e é indispensável que a informação seja rigorosa, clara e adequada.

Nesse sentido, surge a necessidade de grupos multifacetados que ao mesmo tempo que produzem boa informação, também o pensem de forma apelativa e coerente. *There is no inherent reason that computers should use a predominantly visual interface, except that people now prefer it this way (Mirzoeff, 1999, 6).*

Assim, este relatório, para além de demonstrar o trabalho realizado na *Sketchpixel*, pretende também dar a perceber como é fundamental o trabalho de equipa no que respeita à produção de conteúdos digitais, sendo esse um dos focos do mestrado. É importante que grupos tão distintos como criativos e programadores se aproximem de modo a que o seu conteúdo se destaque entre toda a informação difundida.

O presente relatório encontra-se organizado em oito capítulos. No segundo, correspondente à apresentação da empresa, são explicadas as circunstâncias e os motivos que levaram à realização do estágio, os objetivos a alcançar e a contextualização do trabalho desenvolvido.

No capítulo três, ‘Estado da Arte’, é apresentada uma breve explicação das tecnologias utilizadas no decorrer dos diferentes projetos, também com o intuito de justificar a sua utilização, importância e desenvolvimento.

Os três capítulos seguintes correspondem a cada trabalho efetuado na *Sketchpixel*. Todos eles têm uma breve introdução com o objetivo final do projeto. Indica-se seguidamente a parte sobre a qual incidiu o trabalho desenvolvido, visto serem sempre realizados em equipa, e as tecnologias utilizadas. Seguidamente aponta-se o processo de desenvolvimento e por fim algumas conclusões.

O capítulo sete, *WORLD OF DISCOVERIES – ‘CEUTA ONTEM. CEUTA HOJE’*., por se tratar de uma exposição para a qual foram realizados mais do que um trabalho, são apresentados cada um individualmente. No entanto, é feita uma breve introdução à exposição, a sua importância e os novos desafios que trouxe ao trabalho realizado durante o estágio.

Por último, no capítulo oito, são reunidas e apresentadas as conclusões do estágio e um balanço final de toda a experiência.

2 | *Sketchpixel*

A *Sketchpixel* é uma empresa com sede em Braga que surgiu em 2008 e que assume a inovação como forma de agir. Pretende que as novas tecnologias façam parte do quotidiano da sociedade, sempre com o máximo de qualidade e rigor. Aposta num processo baseado na investigação e desenvolvimento tecnológico, com competências multidisciplinares. Desenvolve trabalho em áreas tão distintas como 3D, arquitetura, publicidade, *design*, vídeo e *motion*. Na área das novas tecnologias, a *Sketchpixel* já conta com projetos de realidade aumentada, pirâmides holográficas, vídeo *mapping*, aplicações móveis, soluções *multitouch* e *QR Code*.

2.1 Relação com a empresa

Mais recentemente, em 2014, assinou um protocolo de cooperação com o Instituto Politécnico de Tomar, na sequência do qual resultou a criação do Laboratório Q207. Este laboratório apoiou o mestrado em Produção de Conteúdos Digitais e permitiu que este estágio se realizasse.

A empresa tem também projeção internacional, com escritórios em França e no Dubai. Ao longo do estágio foram realizados trabalhos para locais tão distintos como Brasil e Mónaco. Desta forma, foi possível a integração de vários alunos do mestrado, que, ao mesmo tempo que mantinha alguma conformidade no grupo de trabalho, também possibilitava uma grande troca de conhecimentos.

2.2 Contextualização e motivação

Consciente do principal objetivo da atividade na empresa - comunicar de forma eficaz e com soluções inovadoras, mantendo, acima de tudo, a qualidade - procurou-se essencialmente o rigor, a honestidade e a persistência nos desafios propostos. A avaliação constante das decisões e a atenção ao pormenor auxiliam a manter o foco nas exigências de cada cliente e, consequentemente, no público-alvo.

Assim sendo, realizar estágio na *Sketchpixel* teve, desde início, três objetivos principais. Com o primeiro, cumprindo os requisitos para a finalização do mestrado em Produção de Conteúdos Digitais, pretende-se registar, argumentar e defender o trabalho desenvolvido. O segundo prende-se com a possibilidade de obter uma primeira experiência profissional numa empresa onde se trabalham diferentes áreas tecnológicas. Aqui, foi permitido aplicar os mais diversos conhecimentos, não só os adquiridos na parte letiva do mestrado, mas também nas áreas do *design* e da multimédia. Por fim, a compreensão do que é o trabalho de equipa e a importância de cada elemento, de forma a responder às necessidades do mercado.

É essencial a existência de pessoas que realizem a ponte entre programadores e criativos. Excetuando os trabalhos descritos nos pontos 7.2. e 7.3., todos os projetos realizados ao longo do estágio foram em equipa e sempre com este objetivo presente. Percebeu-se que os mais diferentes conteúdos podem, e devem, ser complementados com diferentes áreas do conhecimento.

Este novo laboratório de investigação, o Q207, foi uma novidade tanto para nós alunos como para a empresa. Trabalhar à distância com *briefings* fornecidos e com colegas que não conhecíamos pessoalmente, marcou também muito do trabalho desenvolvido.

Para além disso, o desenvolvimento de *websites* e aplicações móveis na íntegra foi realizado pela primeira vez no Q207, de forma que foi necessário criar algum método de trabalho individual e em grupo. A gestão de tempos e a junção de várias ideias simultâneas e sucessivas foram, sem dúvida, o maior desafio no decorrer do estágio.

3 | Estado da arte

Os trabalhos desenvolvidos na *Sketchpixel* durante o estágio compreendem diversas áreas, resultando nos mais diversos conteúdos digitais. Desta forma, neste capítulo serão descritos os diversos temas relacionados com o desenvolvimento e evolução dos projetos.

Numa primeira abordagem são apresentadas as tecnologias consideradas fundamentais para o desenvolvimento de aplicações web: o HTML5, CSS3 e JavaScript. São seguidamente analisados os *motion graphics* e a sua importância para a divulgação dos trabalhos e produtos de uma empresa. Por fim, na última secção são postos em confronto as imagens *bitmap* com os vetores, explicadas as suas diferenças e as suas diferentes utilizações.

3.1 HTML5

HTML é a sigla em inglês para *Hypertext Markup Language* que em português significa linguagem para marcação de hipertexto. Esta linguagem surgiu em 1990, inventada por Tim Berners-Lee, com o intuito de partilhar documentos científicos.

Tim Berners-Lee fundou a *World Wide Web* (W3C) em 1994, um consórcio internacional formado por empresas, instituições, investigadores e público em geral com a finalidade de desenvolver a web e tirar partido do seu potencial máximo. Criaram normas e especificações aplicáveis aos diversos setores da web, desde tecnologias e *softwares* até fabricantes e fornecedores (Silva, 2014a, p.26).

A partir desta data foram desenvolvidas novas marcações de HTML e criados novos elementos e atributos. A partir daqui, o HTML deixou de ser apenas uma linguagem de marcação e estruturação de textos e começou a permitir tamanhos, tipos e cores de letras, cores de fundo, texturas e outras estruturações *em total desacordo com o propósito inicial da linguagem* (Silva, 2014a, p.27).

Esta linguagem é também o coração da W3C. Através dela a internet expandiu e amadureceu, tornou-se mais poderosa e aumentou o seu alcance. A própria W3C reconhece a importância do desenvolvimento desta quinta versão do HTML:

The HTML5 Recommendation represents a milestone in the development of HTML but far from being the end of the road and improvements are already well under way. (W3C, 2014a)

O HTML5 veio substituir a anterior, o HTML4, publicado em 1998 pela W3C, que acreditava nessa altura que o futuro seria o XML (*eXtensible Markup Language*). A propósito disso, Tim Berners-Lee afirma em 2006:

Making standards is hard work. Its hard because it involves listening to other people and figuring out what they mean, which means figuring out where they are coming from, how they are using words, and so on. [...]

It is necessary to evolve HTML incrementally. The attempt to get the world to switch to XML, including quotes around attribute values and slashes in empty tags and namespaces, all at once didn't work.

Até esta data, a W3C alegava que não existiriam mais versões de HTML pois a funcionalidades do XML adaptadas ao HTML (que resultou na linguagem XHTML) permitiriam novos caminhos no mundo da web.

A tentativa de evoluir o HTML para conformidade com XML fracassou, em grande parte, devido à sua rigidez de sintaxe. Bastava um erro simples como uma *tag* que não fosse fechada ou umas aspas mal colocadas para que a página nem fosse lida pelo *browser*² (Silva, 2014a, p.47).

A alteração de padrão iniciou-se com o desenvolvimento da linguagem XHTML2.0, no ano 2000, que iria quebrar a compatibilidade com as versões anteriores de HTML e XHTML com a razão de se tornar mais lógica e bem concebida.

Também a publicação em 2003 do *XForms*, uma tecnologia considerada a próxima geração dos formulários para a web, utilizando características de formulários do HTML4, reavivou o interesse pelo desenvolvimento do HTML.

Em 2004, num *workshop* da W3C, a *Mozilla* e a *Opera* apresentaram alguns princípios que fundamentam o HTML5, bem como algumas características relacionadas com os formulários. A proposta foi rejeitada com a justificação de que entrava conflito com a direção previamente escolhida para a evolução da web.

²O termo inglês *browser* é usado na linguagem web para designar um programa capaz de ler e apresentar ao utilizador os conteúdos de um documento escrito em linguagem de marcação. *Browser* vem do verbo *to browse* e foi traduzido para o português como navegador. São exemplos de *browsers* o *Microsoft Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome* e *Apple Safari*, entre outros (Silva, 2014a, p.23)

Com base nisto, estas duas empresas, juntamente com a Apple, formaram a *Web Hypertext Application Technology Working Group* (WHATWG) e anunciaram a sua intenção de continuar a desenvolver as especificações do HTML5. Esta comunidade é liderada por Ian Hickson e mantém as suas convicções a todo o custo.

The WHATWG was based on several core principles, in particular that technologies need to be backwards compatible, that specifications and implementations need to match even if this means changing the specification rather than the implementations, and that specifications need to be detailed enough that implementations can achieve complete interoperability without reverse-engineering each other (W3C, 2014).

Em 2006 a W3C mostrou interesse em participar no desenvolvimento do HTML5 e em 2007 formou um grupo de trabalho para colaborar com a WHATWG na especificação do HTML5. A *Apple*, *Mozilla* e *Opera* permitiram que o W3C publicasse a especificação sob os direitos autorais W3C, mantendo uma versão com uma licença menos restritiva no *site* da WHATWG. O facto desta nova especificação estar a ser também trabalhada por fabricantes de *browsers* igualmente foi importante para o seu desenvolvimento. Esta importância é afirmada por Ian Hickson numa entrevista a Bruce Lawson:

The reality is that the browser vendors have the ultimate veto on everything in the spec, since if they don't implement it, the spec is nothing but a work of fiction. So they have a lot of influence—I don't want to be writing fiction, I want to be writing a spec that documents the actual behaviour of browsers.

Outro dos problemas que surgiam na linguagem HTML é a forma como os *browsers* lidam com a marcação inválida. Este é outro dos temas abordados pela WHATWG para que determinados erros estruturais possam sobreviver ao render dos *browsers*. Este não era um problema para a marcação XML porque os navegadores interrompiam automaticamente a leitura da página assim que encontrassem um erro (Lawson, Sharp, 2011, p.XIII).

Num exemplo simples como:

```
<div><h1>Olá mundo!</div></h1>
```

a leitura da estrutura, devido à mistura de *tags*, pode ser diferente nos diversos *browsers*, tal como a leitura do ficheiro CSS pode resultar numa apresentação completamente desigual. No

interesse de uma boa interoperabilidade, é necessário que todos os *browsers* gerem exatamente a mesma DOM³ quando confrontados com má marcação HTML. Para que isso aconteça, é necessária uma especificação.

Desta forma, as vantagens trazidas pelos documentos desenvolvidos em HTML5 são as seguintes:

- Melhoria da acessibilidade e interoperabilidade;
- Compatibilidade máxima com as versões anteriores;
- Linguagem notoriamente semântica;
- Melhoria na procura de conteúdos (SEO – *Search Engine Optimization*).

Foram introduzidos 28 novos elementos semânticos com o HTML5, muitos deles inspirados em nomes de *class* e *id* atribuídos pelos programadores que quase se tornaram *standard* no HTML4 (Lawson, Sharp, 2011, p.7). Esta nova estruturação das páginas é simples e contribui para uma melhor interpretação das mesmas, tanto por parte dos programadores como dos próprios navegadores. Esta sintaxe torna as páginas mais leves e, conseqüentemente, fáceis de carregar e uma maior transparência na comunicação de conteúdos (Figuras 1 e 2).

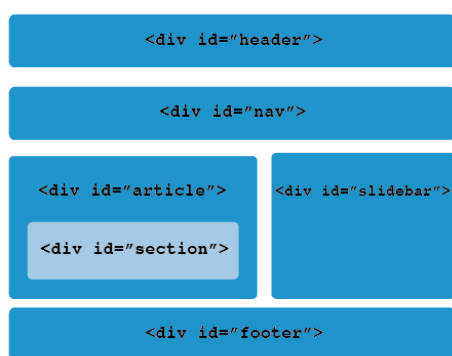


Figura 1: Tags usadas anteriormente com recurso a *div*

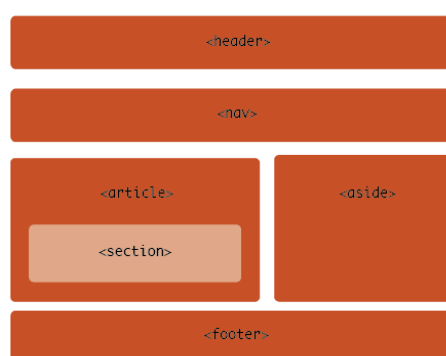


Figura 2: Novos elementos semânticos do HTML5

Outra das vantagens é a facilidade e simplicidade na criação de formulários, que nesta nova versão da linguagem a *tag* <input> reduz a dependência de scripts. Existem novos atributos associados a esta *tag* que permitem a criação de um formulário sólido sem recorrer a JavaScript (Bengtson, 2013, p.396).

³Document Object Model (DOM) é uma plataforma que representa a forma como o conteúdo, estrutura e estilo dos documentos é organizada e lida pelo *browser* (W3C, 2005).

As aplicações web têm vindo a crescer e são muito mais que páginas estáticas de texto, imagens e *links*. Permitem também um padrão livre de reprodução de conteúdos multimédia sem recurso a *plugins*, como *Shockwave*, *RealPlayer*, *Java*, etc., que para além da necessidade de atualizações regulares, podem interferir uns com os outros causando consequências, por vezes graves, na leitura das páginas. Estes *plugins* podem aumentar o tempo de carregamento ou o congelamento do carregamento da página.

HTML5 helps to remedy much of this problem, by introducing a new array of application programming interfaces (APIs) to modern browsers that can be accessed through HTML tags. With HTML5, the browser, which acts as the renderer and interpreter for the markup and code on the Web, can natively provide functions such as video playback and sophisticated animations, eliminating the need for many browser plugins. HTML5 APIs provide a fast and consistent experience while reducing the loading and lag times so characteristic of the old Web (Bengtson, 2013, p.393).

Houve um grande progresso nesta linguagem quando pensamos que surgiu inicialmente apenas para partilhar documentos científicos, sem preocupação com os seus layouts. O HTML5 é então a versão mais recente da linguagem HTML, que evoluiu por oito versões desde a sua primeira especificação de forma a fazer face às necessidades atuais do mundo web, revolucionou a sua evolução, funcionamento e utilização. Esta nova versão é um *standard* que simplificou o trabalho dos programadores e melhorou o acesso a dispositivos e aplicações.

Existe já uma versão do HTML5.1, desenvolvida pela W3C, considerada ainda em fase de rascunho e que virá, possivelmente, substituir a versão 5.

O HTML5 não é apenas a última especificação desta linguagem, estruturada em doze secções, mas um conjunto de tecnologias: linguagem de marcação para acesso e manipulação dos documentos, a CSS que define a apresentação e a aparência do documento e o JavaScript, normalmente utilizado para incluir aplicações específicas de programação de interfaces. Estas tecnologias conferem à web uma capacidade inexplicável que se deve grande parte também à sua universalidade. Esta sua maior vantagem foi também referida por Tim Berners-Lee já em 1997:

The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.

3.2 CSS3

CSS é a abreviação do termo em inglês de *Cascading Style Sheets*, traduzido para português como folhas de estilo em cascata. Por cascata entende-se o seguir/fluir para baixo (Shaw, 2015, p.15).

A utilização da CSS3, em paralelo com o HTML5, permite o desenvolvimento de páginas web com um *design* mais atrativo e agradável. Tal como as novas especificações do HTML5, a CSS3 (ou CSS nível 3) é o último conjunto de especificações pensadas para modelar e formatar os elementos de marcação. Na *homepage* da W3C, comunidade também responsável pelos *standards* CSS, esta linguagem é definida de uma forma clara, simples e precisa:

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.

A principal vantagem dos ficheiros de CSS, para além de ser considerada como uma boa prática a separação entre a estrutura e a apresentação (Shaw, 2015, p.11), é o facto de não sobrecarregar a página HTML. Permite também uma mais fácil manutenção dos websites e que a mesma folha de estilos seja reutilizada. A apresentação do *layout* no dispositivo do utilizador é a função da CSS. Por essa razão, a sua manutenção e atualização devem ser simples e acessíveis (Silva, 2014b, p.31)

O nível anterior existente da CSS era a 2.1 (divulgada em 2002), uma revisão da CSS2, publicada em 1998. Na verdade, o nível 3 da CSS começou a ser realizado nesse mesmo ano. No entanto, e visto que a CSS2 estava a começar a ser implementada pelos *browsers*, a W3C decidiu suspender o trabalho de um novo nível e trabalhar na revisão dessa mesma versão. Apenas em 2005 foram reabertos os módulos da CSS3 e a sua revisão começou (Gasston, 2011, p.2).

O W3C prefere o termo nível em vez de versão, para designar a progressão dos diferentes documentos que especificam as CSS, alegando ser aquele termo mais apropriado. Argumenta informando que cada novo nível é projetado com base no nível anterior, refinando definições e acrescentando novas funcionalidades às já existentes. O conjunto de funcionalidades de níveis anteriores é, sempre, um subconjunto das funcionalidades da versão atual, portanto por ela plenamente suportado. Dessa forma, um agente de usuário que suporte as funcionalidades das CSS de nível atual suporta também todas as funcionalidades das CSS de níveis anteriores (Silva, 2012, p.43).

Na realidade, um dos maiores contributos ao desenvolvimento dos *browsers* tem sido a CSS3. Nos últimos anos, uma vasta gama de navegadores web têm surgido a competir de forma a darem aos utilizadores e programadores as mais recentes tecnologias e até mesmo a

adicionarem e proporem novos recursos. Nesta corrida, o *Internet Explorer* foi ficando para trás devido à sua resistência na implementação da CSS3. Numa altura em que dominava o mercado, não manifestou grande interesse em se manter na vanguarda (Gasston, 2011, p.2).

A abordagem à especificação da CSS3 foi radicalmente alterada em relação às anteriores. Enquanto que estas foram desenvolvidas num único documento, a CSS nível 3 está estruturada em módulos. Dividiu-se a especificação existente por assuntos e criou-se para cada assunto um módulo independente. Alguns desses módulos são revisões da CSS2.1 e outros são elaborados totalmente de raiz.

Visto que existiam diversos pontos a discutir, a ideia de a desenvolver em módulos permite que cada parte possa ser trabalhada por diferentes autores a diferentes ritmos, tal como a sua recomendação. Existem módulos já terminados e recomendados e outros cujas especificações são consideradas ainda experimentais.

Esta divisão *possibilita [também] que fabricantes comecem a implementar, desde já, funcionalidades previstas em módulos mais adiantados no seu desenvolvimento* (Silva, 2012, p.44). É importante, no entanto, notar que não é pelo estado definitivo de qualquer propriedade que a sua aplicação é possível em qualquer *browser*.

O estado de um módulo é definido pela W3C e o seu progresso é indicado através do processo de recomendação em que este se encontra. Quando um documento é proposto e aceite como parte da CSS3, o seu estado é considerado como *Working Draft* (rascunho de trabalho). Essa fase começa com a publicação de um rascunho das especificações para conhecimento e discussão públicos. Um documento pode permanecer neste estado por um longo período de tempo. No entanto, nem todos os módulos passam por este estado. Também pode acontecer que algum módulo já mais avançado passe novamente para esta forma de recomendação (Gasston, 2011, p.3)

Depois disto, o documento passa para *Last Call* (última chamada). Isto significa que o período de revisão terminou e normalmente indica que está pronto a progredir para o próximo nível. É uma última chamada às considerações públicas.

O nível seguinte de um módulo é designado de *Candidate Recommendation* (candidato a recomendação) que significa que a W3C considera que foram resolvidas todas as questões

técnicas, nas últimas revisões não foram encontrados problemas significativos e que os *browsers* podem começar a implementar aquelas propriedade de uma forma definitiva.

Quando um módulo candidato a recomendação é implementado e já demonstra algum avanço sem problemas técnicos, este passa a *Proposed Recommendation* (proposta de recomendação) e é reencaminhado para o Comité Consultivo da W3C. Quando é aprovado por este comité, o módulo passa a *W3C Recommendation* (recomendação da W3C) e significa que está terminado e aprovado por todos os membros da W3C.

É importante para os programadores de CSS conhecerem o funcionamento do processo de normalização e o estado de implementação de cada propriedade. Numa altura em que a CSS3 continua em desenvolvimento, a sua sintaxe pode sofrer alterações ou uma propriedade ser inteiramente descartada.

Alguns módulos ainda se encontram pouco desenvolvidos e confusos. Ao mesmo tempo, há certos recursos que necessitam ser implementados para que possamos ver como funcionam na prática. No entanto, há certas propriedades que estão sujeitas a diferentes interpretações por parte dos fabricantes de *browsers*. Para impedir que o mesmo código apareça de forma diferente nos diferentes navegadores, devido a estas especificações ainda não terminadas, cada um dos diferentes mecanismos de *render* definiu o seu próprio prefixo, um código para acrescentar no início de cada propriedade experimental.

Assim, em qualquer propriedade ainda não recomendada por parte do W3C, é aconselhado que se faça o seguinte:

```
div {
  -moz-propriedade: valor; /* Firefox */
  -ms- propriedade: valor; /* Internet Explorer */
  -o- propriedade: valor; /* Opera */
  -webkit- propriedade: valor; /* WebKit */
}
```

Aqui são mostrados os prefixos para os diferentes *browsers*. Quando se refere *WebKit*, os navegadores implícitos são o *Google Chrome* e o *Safari*. São referidos assim devido à sua utilização simultânea do mesmo mecanismo de *layout* que tem esse nome.

A repetição de código pode parecer dispensável mas é necessária para que todos os navegadores implementem a propriedade de igual forma.

Em muitas referências é sugerido que, para além dos prefixos nas propriedades experimentais da CSS, se utilize também a propriedade sem qualquer prefixo. Esta teoria previne que, caso a propriedade passe a estado definitivo de desenvolvimento, não se tenham de rever todas as folhas de estilo para a adicionar.

```
div {  
  -moz-propriedade: valor; /* Firefox */  
  -ms- propriedade: valor; /* Internet Explorer */  
  -o- propriedade: valor; /* Opera */  
  -webkit- propriedade: valor; /* WebKit */  
  propriedade: valor;  
}
```

Segundo Gasston (2011), esta técnica é correta se mais de um *browser* tiver implementada a propriedade de forma compatível, significando que a recomendação está estável. Caso contrário, e caso ainda se encontre em revisão, a sintaxe está sujeita a alterações e a propriedade sem prefixo pode causar conflitos ou, em casos extremos, não funcionar de todo. Propriedades ainda pouco desenvolvidas como é o caso dos gradientes, que serão apresentados no capítulo 5, Gausston (2011) não vê necessidade da utilização da propriedade sem prefixo.

Outra boa prática introduzida na folha de estilos e que permite uma maior uniformização dos *browsers* é a *CSS Reset*. Este conjunto de regras, quando colocadas no ficheiro .css eliminam todas as pré-formatações concebidas pelos *browsers*, que normalmente também variam por cada fabricante. Para ultrapassar determinadas discrepâncias no que respeita a elementos como margens, *padding*s, listas, tabelas, etc., Eric Meyer propôs, em 2007, um conjunto de regras que uniformizam os estilos de elementos que têm, à priori, alguma formatação por parte dos fabricantes dos *browsers*.

We think of our CSS as modifying the default look of a document—but with a “reset” style sheet, we can make that default look more consistent across browsers, and thus spend less time fighting with browser defaults (Meyer, 2007).

Este é o propósito da *CSS Reset*: colocar intencionalmente um estilo padrão generalizado em todos os *browsers*. Existem outras propostas de *CSS Reset*, no entanto, a de Meyer é a mais utilizada. Existe ainda a *Hard Reset*, sugerida em 2004 por Andrew Krespanis que, com

menos linhas de código e através do seletor universal (*) resolve o problema que respeita às margens e *padding*s, no entanto não é tão completa nem tão aconselhada:

Today, it's recognized that using the universal selector has repercussions on web page performance because of the resource tax involved in selecting and assigning styles to all elements (Tuck, 2010).

Existe ainda um módulo de CSS3 que permite não só conferir aspecto gráfico aos *websites* mas que possibilita criar um *layout* sensível ao tamanho do dispositivo do utilizador. As *Media Queries* são um dos recursos mais revolucionários de todas as especificações da CSS e é um dos assuntos mais debatidos nas mais diversas conferências, com o desejo de superar as limitações de navegar na web em dispositivos menores (Gasston, 2011, p.21).

Enquanto se acedia à *World Wide Web* apenas através de computadores, fixos ou portáteis, escrever CSS era bastante simples. Existem as questões dos diferentes *browsers* e plataformas, como visto anteriormente, mas que facilmente eram reduzidas. Porém, nos últimos anos houve uma explosão de novos dispositivos e apresentar o conteúdo de um *website* da mesma forma em todos eles não mais faz sentido.

O módulo de *Media Queries* encontra-se no estado de recomendação da W3C. Para além de se encontrar num estado final, a sua implementação já está disponível em todos os navegadores mais recentes.

Esta propriedade prevê uma sintaxe que permite apresentar determinados estilos de forma adequada ao dispositivo com base na altura e largura da janela do *browser* do dispositivo, na sua orientação e até resolução. Conferem uma grande flexibilidade aos estilos para que seja possível alterar toda a estrutura de um documento HTML com base apenas no tamanho do dispositivo. Um exemplo da aplicação prática desta regra, utilizando por exemplo a largura do dispositivo é a seguinte:

```
@media screen and (min-width: 400px) {  
  h1 {  
    propriedade: valor;  
  }  
}
```

Ethan Marcotte, em 2010, publicou um artigo na sequência da explosão dos dispositivos móveis, no qual refere, pela primeira vez, o conceito de *Responsive Web Design* (RWD). Nesse artigo, considera que o trabalho desenvolvido para a web, ao contrário da arquitetura que tem um carácter permanente, é transitório.

Marcotte considera ainda que, para além das *media queries*, para se criar um *layout responsive*, é necessária uma grelha fluida e imagens flexíveis. Neste caso, é também importante que as medidas da página não sejam fixas mas em percentagem visto que desta forma o *layout* mais facilmente se adequa ao tamanho do ecrã (Silva, 2014, p.35).

As *media queries* são de simples utilização e evitam o recurso a JavaScript para resolver os problemas de redimensionamento de elementos simples como imagens e textos. Para além disso, os *scripts* necessitavam de atualização permanente de cada nova versão de *browser* que surgisse (Gausston, 2011, p.10).

Esta nova versão da CSS traz consigo um vasto conjunto de tecnologias e extensões que permitem uma maior flexibilidade e controlo no *layout* das páginas. Ela possibilita, pelas razões apresentadas anteriormente e outras que serão abordadas em capítulos seguintes, que os *designers* pensem grande sem que questões técnicas lhes cortem a criatividade.

Muitos dos módulos da CSS3 ainda não estão num estado definitivo e novas propriedades podem ainda surgir, tal como outras, que ainda usamos em versão experimental, podem desaparecer, como visto anteriormente. O W3C está sempre atento aos debates e conferências de forma a poder melhorar o desenvolvimento desta linguagem (Gausston, 2011, p.229).

A CSS3 está a mover-se no sentido de ser mais do que uma linguagem de apresentação descritiva. Com as suas propriedades de cálculo, *media queries*, animações e transições, por exemplo, começa a aproximar-se de linguagens como JavaScript e PHP. Ela permite *layouts* cada vez mais dinâmicos e flexíveis apenas com programação do lado do cliente. Também a introdução de constantes e variáveis na sua sintaxe está a ser pensada e proposta, apesar deste tema gerar algumas controvérsias.

Embora o HTML5 tenha trazido importantes recursos para a web, a CSS3 implementou propriedades com grande impacto visual como animações a duas e três dimensões, sombras, cantos redondos, gradientes, etc. Consegue, neste momento, alterar muitos mais que cores,

fontes e espaçamentos, aquilo para que foi criada inicialmente.

3.3 JavaScript

O JavaScript foi criado pela *Netscape* em parceria com a *Sun Microsystems* com o objetivo de acrescentar interatividade a uma página web. A primeira versão, denominada JavaScript 1.0, foi lançada em 1995 e implementada em março de 1996 no *browser Netscape Navigator 2.0*. (Silva, 2010, p.27).

Esta linguagem de programação tem sido uma tecnologia-chave para o desenvolvimento web. Há quem veja o JavaScript apenas como uma forma de adicionar animações e alguns eventos ao HTML. No entanto, possibilita também manipular eventos para objetos na página, mostrando e ocultando conteúdos, recolher informações a partir de uma fonte *online* e exibi-las ou até cumprir funções de normalização de dados e visualização (Mitchell, 2010, p.454).

Não existe, por exemplo, nenhuma forma de processar os dados de um formulário HTML com a utilização apenas de elementos de marcação. Com o HTML é possível apenas criar os campos de um formulário para serem preenchidos pelo utilizador. Com ele não é possível que a recolha de dados seja enviada para o servidor.

JavaScript é uma linguagem desenvolvida para o lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no *browser* do utilizador. Isto significa que os *scripts*⁴ são executados pelo *browser* aquando do carregamento da página ou sempre que um evento seja acionado pelo utilizador (W3C, 2014c).

Com o JavaScript, é possível escrever marcação HTML e introduzi-la num ficheiro existente. Ele é capaz de definir, alterar e controlar de forma dinâmica um documento HTML, como aspectos relacionados com cor de fundo, de textos e de *links*, e até mesmo interferir no posicionamento dos elementos. Possibilita ainda manipular a folha de estilos associada ao documento criando novas regras CSS ou anulando outras existentes (Silva, 2010, p.24).

Com todas estas capacidades, que geram muitas vezes comportamentos inesperados e não solicitados por parte do utilizador, esta linguagem nem sempre foi bem recebida. O facto de

⁴Um *script* é um programa escrito em JavaScript que não necessita de pré-processamento antes de ser executado. É integrado no documento HTML através da *tag* `<script>` e permite efetuar uma ponte entre o *browser* e a plataforma (W3C, 2014c).

criar janelas *pop-up*, alterar as dimensões do *browser*, retirar menus, fechar e abrir janelas levou à criação de princípios básicos para a sua utilização, em conformidade com os Padrões Web (Silva, 2010, p.24).

Estes padrões não são um *standard* nem uma recomendação como existem para o HTML e a CSS. São apenas princípios que gerem a sua utilização que defendem a não obstrução das páginas, incutindo a sua utilização apenas para melhoria da plataforma. Isto significa que a utilização de JavaScript não deve comprometer o acesso aos conteúdos, voltando-se para o enriquecimento da experiência do utilizador sem criar barreiras de acessibilidade e usabilidade (Silva, 2010, p.26).

3.3.1 *Frameworks*

[...]A set of tools, libraries, conventions, and best practices that attempt to abstract routine tasks into generic modules that can be reused (Croft, 2007).

Esta é uma definição básica das *frameworks* JavaScript que, no fundo, agilizam o processo de desenvolvimento e contêm um conjunto de funcionalidades que são muitas vezes usadas, permitindo aos programadores web focarem-se nas características específicas da plataforma e não reinventar código para tarefas mais generalistas.

Existem cada vez mais *frameworks* que simplificam em muito a criação de *websites*. São normalmente acompanhadas de documentação que explica todas as suas possibilidades e todo o processo necessário para realizar determinada tarefa. Desta forma, uma *framework* é composta por classes e/ou métodos que são um conjunto de instruções pré-definidas. As principais razões para a sua utilização são:

- Minimização de complexidade;
- Agilidade no processo de criação;
- Código mais estruturado.

3.4 *Motion Graphics*

Motion Graphics é o termo usado para a combinação entre palavras, imagens, gráficos, som e vídeo. Ou seja, pode dizer-se que combina a animação com filme, o *design* com tipografia. É um termo atualmente muito utilizado e pode ser aplicado nas mais diversas situações.

Este termo vem claramente das colagens que a partir do século XX, com as novas vanguardas, foram incorporadas nas diferentes linguagens artísticas. Desde o cubismo ao dadaísmo, passando pelo futurismo e surrealismo, todas estas frentes viram na colagem um novo recurso, uma possibilidade de experimentação e confronto com as formas de arte tradicionais (Vargas, Souza, 2011, p.54).

A ilusão de movimento com colagens e a sua manipulação através *softwares* de animação tornaram-se muito comuns no campo audiovisual. Desde a expansão dos meios de comunicação de massa no século XX, o *design* tem-se imposto em função das tecnologias de artes gráficas, vindo de origens diversas como estes processos básicos de composição.

Apesar de muito utilizados em toda a histórias da arte, o termo *motion graphics* democratizou-se no ano de 1960 quando John Whitney, pioneiro da computação gráfica, fundou a empresa *Motion Graphics Incorporated*. Para ele, o computador pode também ser *considerado um meio viável de produção artística na animação e nos filmes* (Vargas, Souza, 2011, p.65).

Desta forma, os *motion graphics*, através das ferramentas digitais, herdaram e modernizaram o espaço da fotomontagem e da colagem, transferindo-o para ambientes de criação mais poderosos e com novas possibilidades técnicas.

À parte da sua história e do seu aparecimento, os *motion graphics* são hoje um meio publicitário muito forte e muito utilizado nas mais diversas plataformas. Muitas empresas incluem no seu *website* uma animação em *motion graphics* com o intuito de cativar mais clientes e de vender melhor o seu produto.

As animações em *motion graphics* passaram, em pouco tempo, a ser indispensáveis na apresentação de uma empresa. Os vídeos tornaram-se no meio preferido para a visualização de conteúdos online e revelaram-se na melhor forma para profissionais de marketing comunicarem suas mensagens.

In 2014, online videos are no longer a 'nice-to-have' marketing tool, but a 'must-have'. With animation viewers retaining 58% more information than viewing just images with black and white text (Blagden, 2014).

De entre as vantagens que possuem, os *motion graphics* destacam-se pelo aumento da acessibilidade. Eles permitem que informações mais complexas sejam passadas de forma mais acessível e envolvente. Desta forma, permitem condensar os conteúdos em algo mais cativante, tornando-se mais simples de absorver, compreender e partilhar (Blagden, 2014).

Também as pesquisas feitas através de motores de busca estão a alterar-se no sentido em que, nos seus resultados, apresentam cada vez mais vídeos em vez de páginas estáticas. Isto deve-se não só ao formato de apresentação mas também à sua forma simplificada de apresentar mensagens-chave, ideias e aumentar a consciência do produto ou serviço para um público mais vasto.

According to Internet Retailer, online shoppers are 85% more likely to purchase a product after watching a product video. Online retailers have tested and proven for themselves that when someone watches their online videos, they're more likely to buy, spend more time on the site and increase their order size (Blagden, 2014).

Devido a todos estes estudos e alterações de paradigmas na sociedade, os *motion graphics* apresentam-se como uma ótima solução de *marketing*: possibilitam a criação de uma história e criam interação e envolvimento num curto espaço de tempo.

É comum definir-se *motion graphics* como *design* gráfico em movimento, pela sua capacidade de misturar e manipular além dos elementos gráficos, como a tipografia ou ilustrações. Mas a sua capacidade vai muito além da sua possibilidade imagética e tornaram-se numa ferramenta eficaz na difusão de conteúdos e produtos.

3.5 *Bitmap vs. Vector*

Este tema opõe muitas vezes dois *softwares* gráficos muito populares: *Adobe Illustrator* e *Adobe Photoshop*. Visto que no seu aspeto geral são muito semelhantes, por vezes não são utilizados corretamente para o trabalho pretendido.

Uma imagem *bitmap*, ou mapa de *bits*, consiste no armazenamento de valores de cores em *bits* que compõem cada pixel de uma imagem. Desta forma, cada *bitmap* corresponde a uma matriz de pixéis que se dispõem através de colunas e linhas. A cada pixel é atribuído uma localização e cor específicas (Evening, 2015)

Estas imagens têm a vantagem de poder ser trabalhadas pixel a pixel, dando grande controle sobre os projetos. No entanto, a imagem deve ser mantida no seu tamanho original visto que os pixels não são redimensionáveis e esta contém um número fixo de pixels.

Ao reduzir uma imagem *bitmap* são reduzidos também o número de pixels, causando distorção na imagem. Por outro lado, se uma imagem for aumentada, são adicionados pixels, baseados nos seus 'vizinhos'. Assim, ao ser ampliada, uma imagem *bitmap* perde a sua definição, começando a ver-se a olho nu os pixels que a constituem (Ya-Mei, Jun-Jie, p.75).

Ao contrário das imagens *bitmap*, uma imagem em vetores é composta por linhas e curvas que definem uma fórmula matemática, descrevendo a imagem de acordo com as suas características geométricas (Evening, 2015). Os vetores são também designados como imagens orientadas a objetos por serem facilmente manipuladas através dos vários elementos que os compõem. Cada componente pode ser movido e alterado sem que isso afete outros constituintes da mesma imagem (Ya-Mei, Jun-Jie, p.76).

A grande vantagem dos vetores é a sua possibilidade de redimensionamento. Um vetor é definido pelos seus atributos, como direção, módulo e sentido. Desta forma, sempre que o tamanho de uma imagem em vetores é alterado, a imagem é redesenhada segundo a sua fórmula matemática e o resultado mantém-se tão bom quanto o original.

Os vetores são, contudo, limitados nos detalhes e nos efeitos. Eles possibilitam a criação de gradientes (lineares ou radiais) e regiões de cores uniformes mas nunca terão tanto detalhe nem variações de cor complexas quanto uma imagem *bitmap* onde cada pixel pode ter a sua própria cor. Também efeitos como sombras ou *blur* não são possíveis neste tipo de imagem.

O *Adobe Photoshop* é um programa de edição de imagem usado na maioria das vezes para fotografias e gráficos *bitmap*. Contém inúmeros filtros e efeitos que, quando aplicados a imagens, melhoram aspetos como cor, definição, contraste, saturação, etc. Embora também contenha a possibilidade de criação de formas, estas serão sempre compostas por pixels com uma cor e posição fixas na imagem, não permitindo a alteração ou movimentação de elementos individuais (Ya-Mei, Jun-Jie, p.75).

Contrariamente, o *Adobe Illustrator* é um *software* baseado em vetores para a criação de ilustrações. Isto significa que cada linha contém pelo menos dois pontos ligados entre si

onde é criada uma fórmula matemática em vez de uma linha de pixéis. Oferece, assim, a capacidade de uma edição mais geométrica e independência de resolução das imagens, podendo ser exportadas para vários formatos sem nunca existir perda de definição (Ya-Mei, Jun-Jie, p.76).

Desta forma, é importante desde início conhecer bem o tipo de trabalho que se vai desenvolver para que o *software* corresponda às necessidades de cada projeto. A semelhança entre o *layout* do *Adobe Photoshop* e *Adobe Illustrator* e o seu sistema de *layers* levam muitas vezes a dúvidas na hora da decisão. Nem sempre é possível que apenas um destes *softwares* realize todas as tarefas pretendidas. Contudo, é essencial conhecer os seus princípios e propósitos para que o trabalho seja bem sucedido.

4 | XPhoto

Este projeto consiste no desenvolvimento de uma aplicação *iOS* e *Android* e de um *website* onde se criam eventos e se partilham fotografias sobre os mesmos. É uma plataforma na qual se juntam o *Instagram* com a funcionalidade de eventos do *Facebook*, com a particularidade de os eventos poderem ser divididos por categorias e com dados de GPS. Estas categorias permitem que cada evento seja associado a uma palavra-chave, criando assim uma rede de interesses entre os utilizadores, para além de manter as fotos mais organizadas.

Cada evento poderá ter vários participantes, e cada participante pode partilhar as suas fotos com outros do mesmo evento. Caso o utilizador seja dono do evento ou um dos seus gestores, é-lhe permitido enviar convites e definir informações como data, nome, descrição, localização, entre outras.

4.1 Tecnologias Utilizadas

O trabalho desenvolvido neste projeto incidiu sobre o *website*, utilizando o HTML versão 5 e o CSS (*Cascading Style Sheets*) nível 3 como tecnologias para a sua produção. Foi trabalhado apenas o código do lado do cliente, ou seja a parte *front-end* da plataforma.

4.2 Processo de Desenvolvimento

Existia neste projeto uma equipa, cada um responsável por cada parte. Foi dividido em três fases: a primeira de arquitetura de informação e *design*, a segunda parte para desenvolvimento da plataforma para o lado do cliente e a terceira seria o *back-end*, recorrendo a ASP.NET com MySQL.

Após terem sido estruturadas e desenhadas as primeiras páginas, o seu desenvolvimento teve início, também para que o cliente pudesse ter uma primeira percepção de como ficaria o *website*.

Nesta segunda fase, era importante que o *layout* ficasse o mais próximo daquilo que o cliente tinha aprovado na fase anterior, o *design*, e que alguma interação já existisse para

que determinadas funcionalidades pudessem ser testadas. Para que isso fosse possível, a CSS3 foi uma tecnologia essencial. Muitas das suas novas propriedades são indispensáveis na apresentação desta plataforma.

A primeira é a utilização da propriedade de fontes que não apenas as que são do sistema, uma lista que anteriormente era restrita mas que garantia o seu correto funcionamento. Neste projeto, para incluir uma fonte que não as pré-definidas, foi utilizado o diretório de fontes da Google (<https://www.google.com/fonts>). Através desta ferramenta é permitido que um sem número de fontes sejam incluídas em *websites*.

Com o *Google Fonts*, é possível procurar, filtrar e comparar qualquer uma das fontes na sua lista. É também possível usá-la ou adicioná-la a uma coleção ou ver exemplos concretos da sua utilização.

Depois se selecionada a fonte desejada, é apresentada uma página que exhibe todos os pesos existentes desse mesmo tipo de letra (Figura 3).

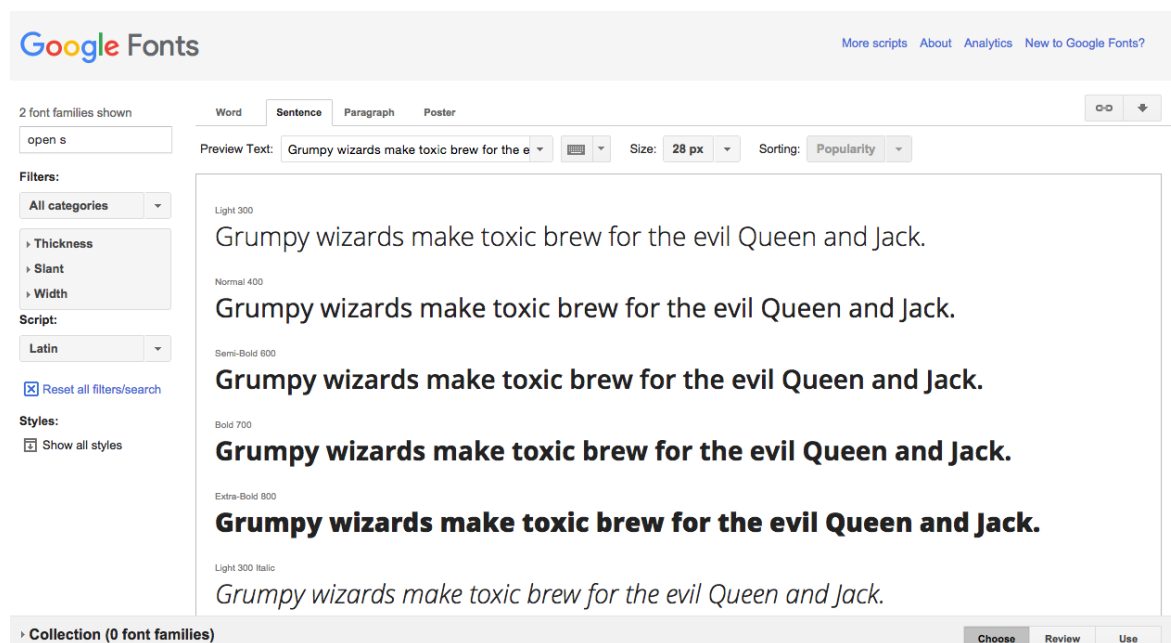


Figura 3: Lista de pesos existente para a fonte *Open Sans* (fonte utilizada no projeto)

Se optarmos pela sua utilização, através do botão *Quick Use* têm-se todas as informações necessárias para a sua utilização como o peso que esta fonte vai ter no *site* (quantos mais estilos forem selecionados, mais impacto a fonte vai ter no carregamento), o código para inserir no documento HTML e a sua forma de integração no ficheiro CSS (Figura 4).

Google Fonts More scripts About Analytics New to Google Fonts?

1. Choose the styles you want:


Open Sans

- Light 300
- Light 300 *Italic*
- Normal 400
- Normal 400 *Italic*
- Semi-Bold 600
- Semi-Bold 600 *Italic*
- Bold 700
- Bold 700 *Italic*
- Extra-Bold 800
- Extra-Bold 800 *Italic*

2. Choose the character sets you want:

Greek (greek) Greek Extended (greek-ext) Latin (latin) Vietnamese (vietnamese) Cyrillic Extended (cyrillic-ext)

Latin Extended (latin-ext) Cyrillic (cyrillic)



Page Load
15

Impact on page load time

Tip: Using many font styles can slow down your webpage, so only select the font styles that you actually need on your webpage.

3. Add this code to your website:

```
<link href='https://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
```

Instructions: To embed your Collection into your web page, copy the code as the first element in the <head> of your HTML document.

[» See an example](#)

4. Integrate the fonts into your CSS:

The Google Fonts API will generate the necessary browser-specific CSS to use the fonts. All you need to do is add the font name to your CSS styles. For example:


```
font-family: 'Open Sans', sans-serif;
```

Instructions: Add the font name to your CSS styles just as you'd do normally with any other font.

Example:

```
h1 { font-family: 'Metropolis', Arial, serif; font-weight: 400; }
```

Google Fonts are powered by the



Check out more [advanced techniques for using Google Fonts](#) like the [WebFont Loader](#) — to fight the flash of unstyled text!

Quick use: Open Sans
Add "Open Sans" to your collection

Figura 4: Instruções para a utilização de fontes

A inclusão deste método já tinha sido feita na CSS2, no entanto, por falta de implementação, foi retirado na versão 2.1. A nova geração de *browsers* tem mostrado bastante interesse neste tema e melhorou esta opção, dando novas possibilidades ao trabalho dos *web designers* (Gausston, 2011, p.49).

A introdução da propriedade `border-radius`, resultando em caixas com cantos redondos, foi talvez uma das maiores vitórias no mundo da CSS. Até à CSS2, tudo na web era quadrado e paralelo. Arredondar cantos, que é agora tão simples, requeria algumas soluções incrivelmente complexas.

Desde o início da web são utilizados cantos redondos em elementos das páginas. No entanto, nunca houve nenhuma formas de criá-los sem recurso a imagens. Para criar uma caixa de largura flexível com quatro cantos redondos significava a criação de quatro imagens que eram adicionadas à linguagem de marcação com pelo menos dois elementos não-semânticas, o que tornava a manutenção do *website* mais difícil (Gausston, 2011, p.108). Por vezes, um pixel poderia fazer a diferença na colocação destas imagens.

Os módulo *Backgrounds and Borders*, neste momento candidato a recomendação, introduz uma forma de arredondar os cantos apenas com a utilização de uma propriedade CSS. Cada canto é tratado como um quarto de elipse, que é definida por uma curva que pode ser regular (circunferência) ou irregular (oval) (W3C, 2014b).

A sua sintaxe é geralmente a seguinte:

```
div { border-radius: valor; }
```

O valor utilizado é normalmente em pixéis, admitindo até quatro valores. Estes valores correspondem a cada canto, começando pelo canto superior esquerdo e seguindo no sentido dos ponteiros do relógio.

Para este projeto foi utilizada esta propriedade em algumas caixas, sempre apenas com um valor, como mostrado no exemplo acima e a Figura 5 apresenta um dos resultados obtido com este novo atributo da CSS3.



Figura 5: Utilização da propriedade `border-radius` na `tag div`

Ainda neste módulo é abordada também a possibilidade de aplicar fundos aos elementos de marcação. Devido à afluência de pedidos nesta área por parte dos *web designers*, muitas das suas novas propriedades já foram implementadas pelos *browsers*, estando assim bastante desenvolvidas e possíveis para utilização.

Ao contrário das versões anteriores da CSS, o nível 3 permite que múltiplas imagens sejam aplicadas aos elementos e até que sejam redimensionadas. Apenas estes dois novos recursos já representam uma grande inovação mas a especificação vai mais longe e possibilita também o controlo no posicionamento da imagem.

```
div {  
    background-image: url('');  
    background-position: valor;  
    background-repeat: valor;  
}
```

Estas novas propriedades permitem criar elementos como o que é mostrado na Figura 6, na qual o triângulo que se encontra no lado direito é uma imagem definida como fundo, sem repetição e posicionada a 97% do final da caixa.



Figura 6: Exemplo da utilização de uma caixa com imagem de fundo e com a sua repetição e posição definidas

Um dos maiores desafios na construção destes *layouts* foi na utilização do pseudo-seletor `:checked`. Os pseudo-seletores são uma extensão de um seletor já existente no documento de marcação e não são elementos independentes. Modificam uma regra existente, dando-lhe um cenário específico quando é ativado, permitindo assim um controlo mais preciso sobre o HTML (Shaw, 2015, p.45).

No caso particular do XPhoto, para criar *checkboxes* estilizadas, foi necessária a combinação de um `input` do tipo `checkbox` com uma `label` a ele associada. Desta forma, é possível adicionar ou remover conteúdo dinamicamente apenas através do HTML e da CSS. Antes da CSS3, para que isto acontecesse, era necessário recorrer a JavaScript para proceder a alterações do

lado do servidor e redesenhar o documento com o novo conteúdo (Shaw, 2015, p.58).

A utilização deste pseudo-seletor, `:checked`, permite então atribuir dois estados diferentes ao `input` do tipo `checkbox`. Para que isto aconteça, é necessário que o `input` esteja oculto através da propriedade `display:none;`, definida na CSS, e especificar diferentes estilos para a `label`, consoante a seleção ou não do elemento de marcação.

Esta formatação permite criar interatividade nas páginas, sem qualquer recurso a JavaScript e sem estarem associadas a qualquer base de dados. Este processo foi necessário repetir em seis das páginas desenhadas inicialmente, sempre com estilos distintos. No entanto, a base para a criação é a mesma.

Na Figura 7 é mostrado um exemplo da aplicação prática deste método. À esquerda é mostrada a formatação da `label` e à direita a mesma `label` formatada através do pseudo-seletor `:checked`.

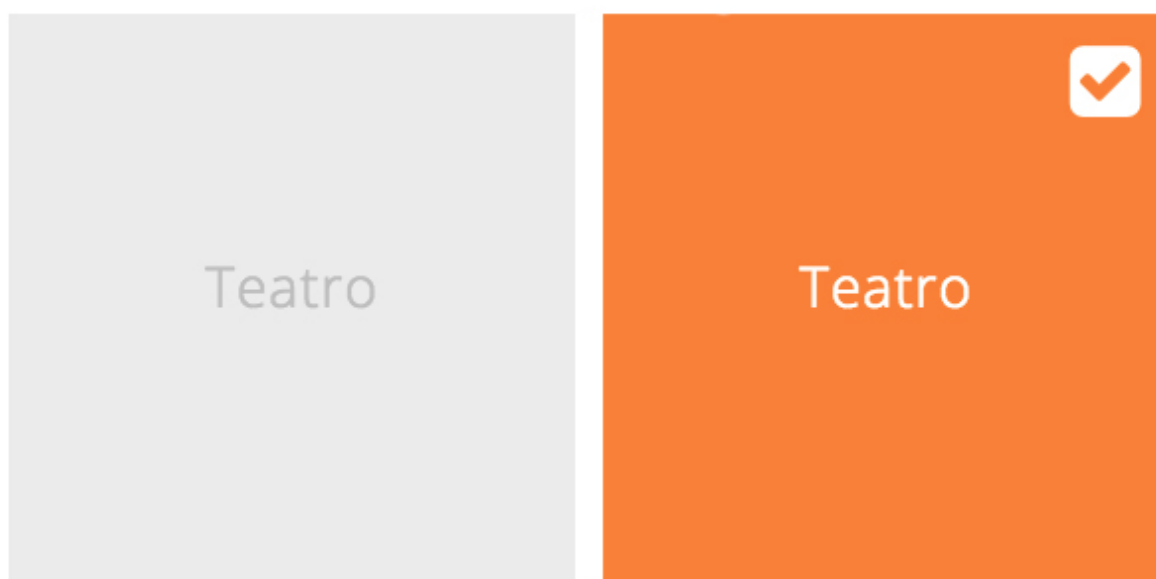


Figura 7: Utilização de uma `label` associada a um `input` para adquirir diferentes estados

4.3 Considerações Finais

Após terem sido terminadas as primeiras páginas e realizadas as primeiras experiências de usabilidade, foi reformulado o *design* por deteção de algumas falhas na estrutura e a necessidade de inclusão de novas funcionalidades.

Foram, por essa razão, criadas novas páginas individualmente para colocar no *website*, estruturadas em conjunto com outro membro da equipa. Foram desenvolvidas mais algumas páginas, umas completamente de novo e outras com alguns elementos já criados na primeira fase do projeto.

São mostradas, nas Figuras 8 e 9, as duas fases desenvolvidas para este projeto no decorrer do mesmo. Depois disto, passado algum tempo, o projeto foi suspenso e por essa razão não foi terminado. À data de 11 de Outubro de 2015 estava uma versão de teste *online* disponível no URL: <http://appxphoto.com/>, com algumas das páginas da última versão produzida.

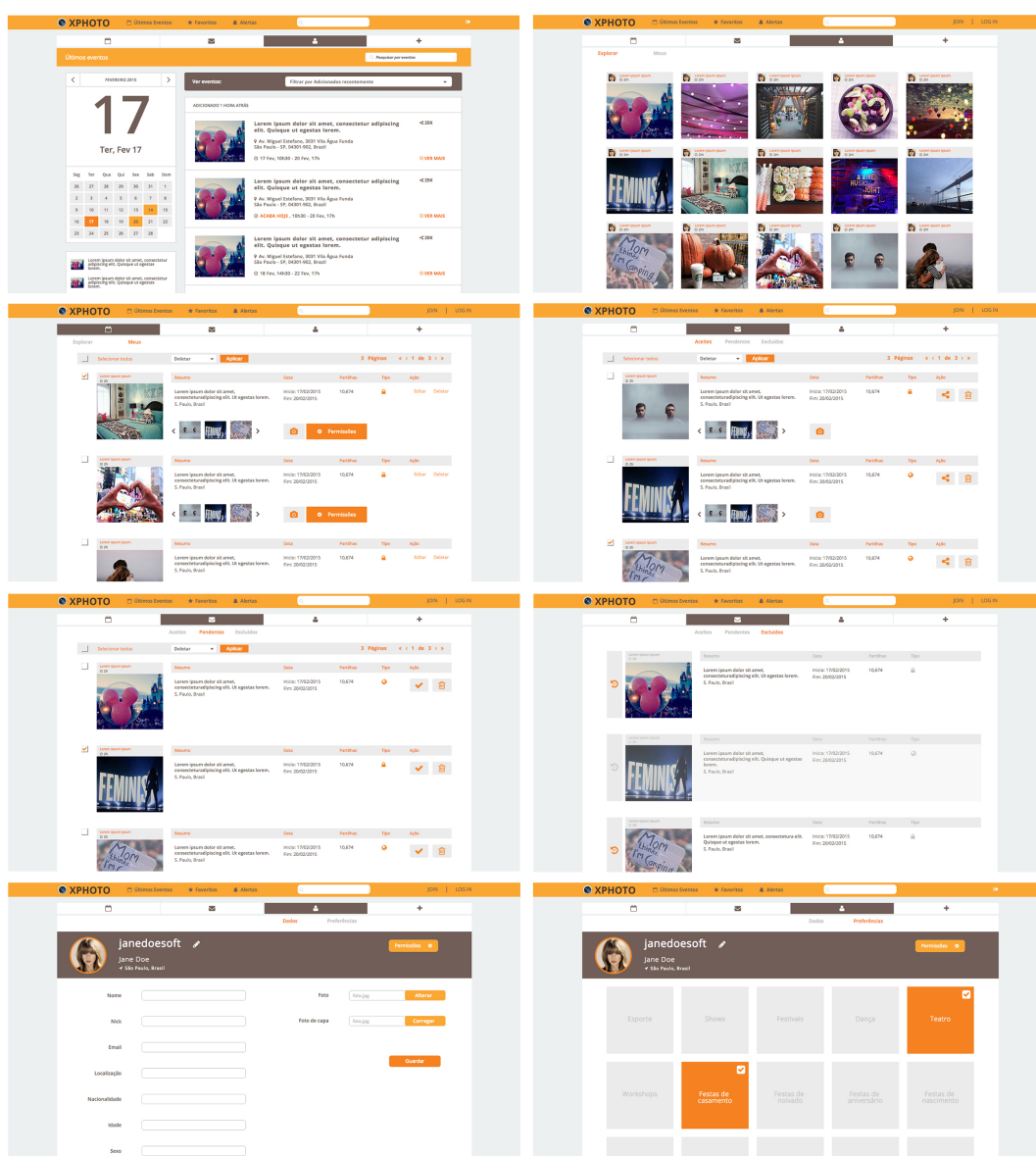


Figura 8: Primeiros layouts do projeto concebidos

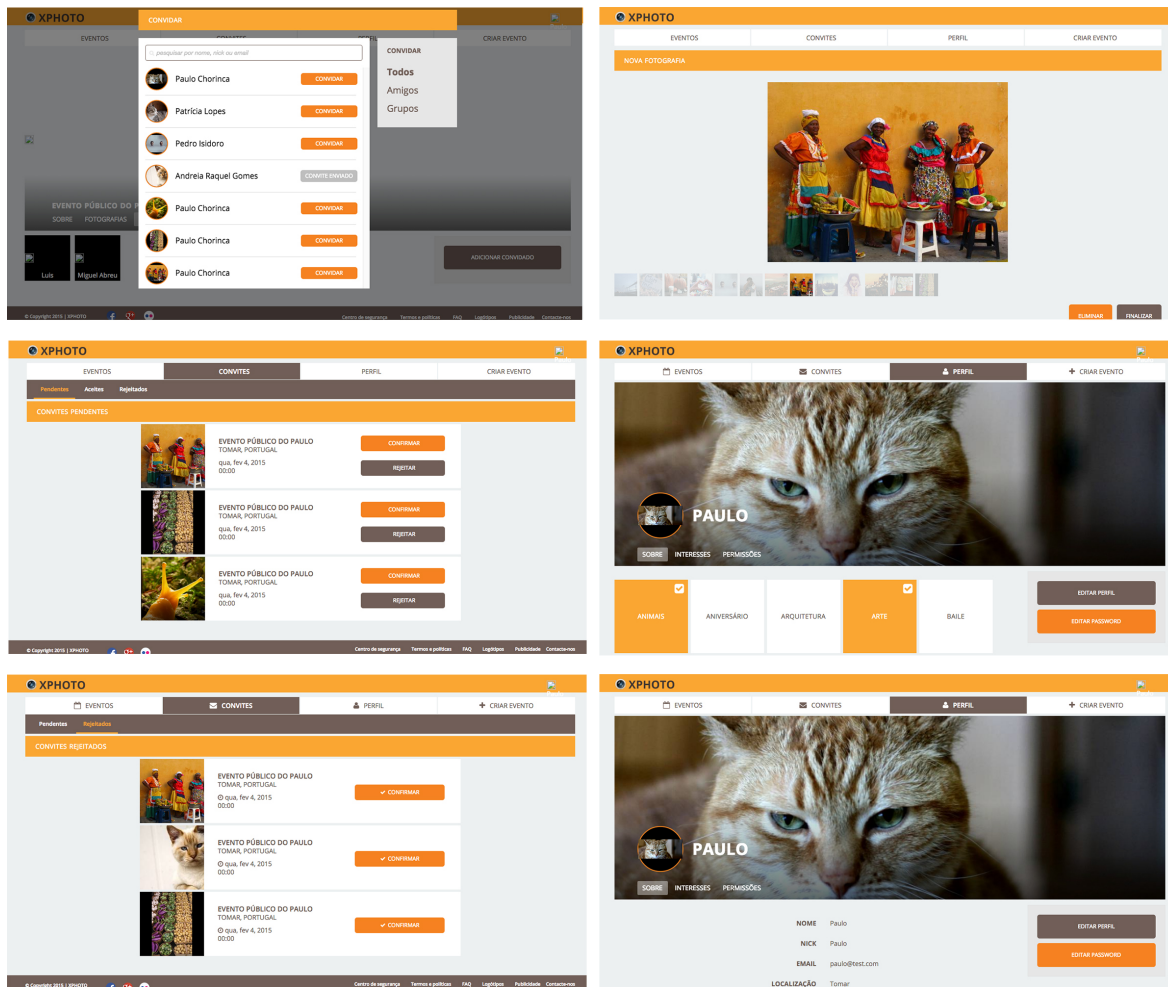


Figura 9: Reformulação do projeto ao nível da estrutura e do *layout*

5 | Aweshome

Este projeto compreende o desenvolvimento de um *website* e de duas aplicações móveis (*iOS* e *Android*). Pretende ser uma forma de aluguer de casas ou quartos e a sua estrutura foi baseada noutros *websites* com a mesma finalidade como é o caso do *AirBnB* e *Housetrip*. A particularidade deste trabalho centra-se na sua localização: estes alugueres são de casas apenas no Mónaco. Tem também informação sobre restaurantes, eventos e outros aspetos sobre a cidade.

5.1 Tecnologias Utilizadas

A parte do trabalho desenvolvida para este projeto incidiu mais uma vez sobre o *website*, utilizando o HTML5 e o CSS3 para a sua estruturação inicial.

5.2 Processo de Desenvolvimento

Para a criação desta plataforma manteve-se a equipa anterior e as fases de trabalho passaram pelas mesmas etapas: iniciou-se pela arquitetura de informação e *design*, passando para a estruturação do *front-ent* e, finalmente, para o *back-end*. Neste trabalho a coordenação entre a equipa foi um ponto muito importante dado que foi um projeto muito grande, com muitas reformulações e o tempo ia sendo cada vez mais escasso.

Numa primeira fase, foi apenas desenhada a página inicial e foi dado início ao seu desenvolvimento que, tal como no projeto anterior, serviria para testar a sua funcionalidade e usabilidade (Figura 10).

Nesta primeira fase é importante destacar a utilização da *tag input* ao nível do HTML5. A redução de dependência de *scripts* e os novos tipos que surgiram, especialmente *data* e *range*, permitiram a esta primeira etapa ter uma melhor perceção da plataforma sem a necessidade de despende muito tempo.

Existiu posteriormente uma grande reformulação ao nível do *design* e uma alteração no método de trabalho. Foram apenas desenvolvidos elementos individuais que estariam presentes

em todas ou quase todas as páginas do *website* (Figura 11).

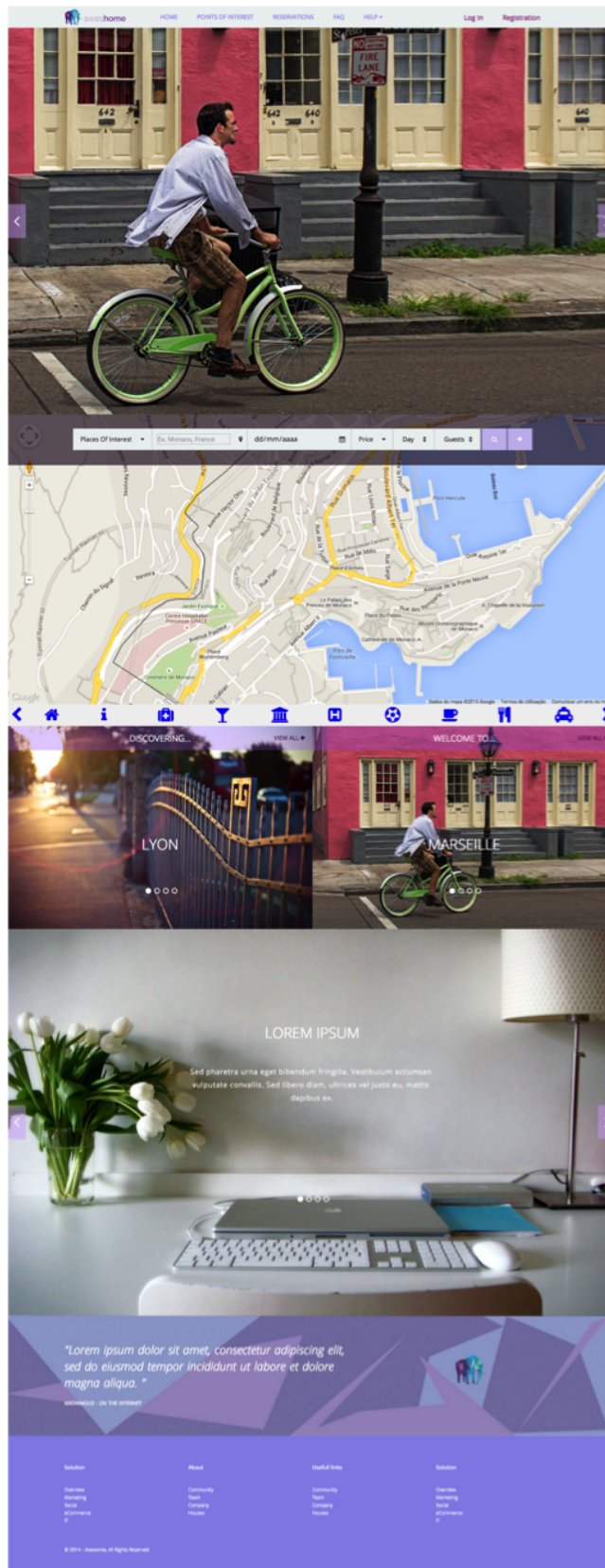


Figura 10: Primeira versão da *homepage* do Aweshome

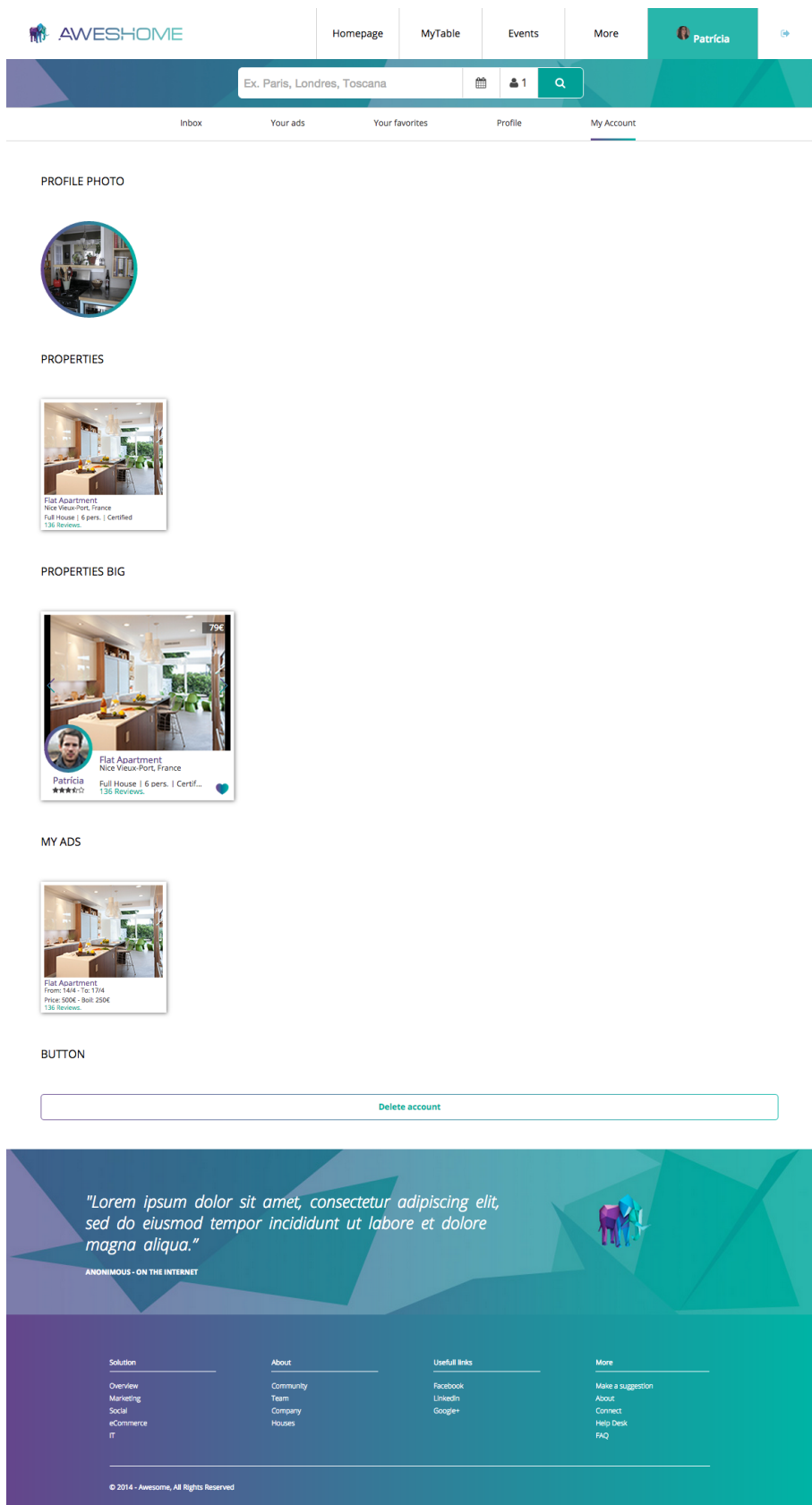


Figura 11: Página de elementos individuais da plataforma Aweshome

O desafio consiste em atribuir-lhes apenas *class* e, quando necessária alguma alteração ao elemento no decorrer da construção do *site*, será atribuído um *id* aos elementos. Este *id* permite controlar um elemento em particular, especificando a hierarquia do ficheiro CSS, dando assim plena utilização à sua natureza em cascata (Shaw, 2015, p.17).

Mais uma vez, a CSS3 foi essencial para a construção destes elementos. Foi necessária especial atenção para os gradientes que em versões anteriores da CSS, tal como outros efeitos gráficos, era necessário exportar uma imagem. No entanto, nesta nova versão é possível criar este tipo de preenchimentos sem usar editores de imagem.

Utilizar gradientes com CSS3 pode parecer ainda um processo complexo devido à sua recente adoção como *standard* por parte do W3C. A sua implementação nos *browsers* está ainda numa fase de transição, exigindo sintaxes bastante diferentes para atingir o mesmo resultado (Shaw, 2011, p.131).

Para a implementação correta desta propriedade, para que todos os navegadores fizessem a sua leitura adequada, foi utilizado um gerador de código *online*, o *Ultimate CSS Gradient Generator*. Esta plataforma possibilita a criação de gradientes através da introdução das cores e o seu sentido, gerando o código para cada *browser* com as suas características específicas. Embora esta ferramenta não permita todas as opções, foi uma ajuda neste caso particular, visto ser um gradiente simples.

Outro elemento que exigiu especial atenção foi uma caixa, correspondente à gestão de anúncios, que contém uma animação e uma transição através do pseudo-seletor `:hover`. Estas novas propriedades, inseridas na CSS3, conferem movimento a uma página estática sem recurso a JavaScript (Figura 12).

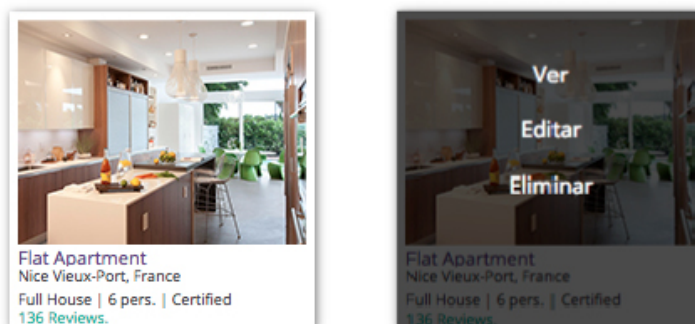


Figura 12: Caixas com a aplicação das propriedades de animação e transição através do pseudo-seletor `:hover`

A diferença entre as transições e as animações é que a primeira é implícita e a segunda é declarada. Isso significa que as transições só têm efeito quando existe uma mudança de valor, enquanto animações são explicitamente executadas quando aplicado a um elemento (Shaw, 2011, p.164).

Estas propriedades levantaram algumas questões a respeito da separação das três camadas que normalmente são consideradas numa página web: o conteúdo (HTML), a apresentação (CSS) e o comportamento (JavaScript). Embora já seja um dos módulos oficiais da W3C, ainda de encontram na fase de rascunho de trabalho e podem vir a sofrer algumas alterações. Apesar disso, já quase todos os *browsers* as implementaram e é segura a sua utilização com os prefixos correspondentes a cada um dos motores de *render*.

Outra questão que estas caixas levantam é o canal *alpha*, introduzido também nesta nova versão da CSS e que foi utilizado para aplicar transparência no elemento animado. Este canal é introduzido no modelo de cores RGBA e a sua sintaxe é a mesma que para o modelo RGB, utilizado na CSS2, acrescentando o valor *Alpha*. Este valor é medido entre 0 e 1 em que 0 é completamente transparente e 1 é totalmente opaco, tendo o 0.5 como meio termo.

```
div { color: rgba(red, green, blue, alpha); }
```

Esta propriedade distingue-se da opacidade visto que o canal *alpha* mede a transparência da cor, enquanto a opacidade mede a transparência de um elemento (Shaw, 2011, p.123).

Ainda nestas caixas foi aplicada uma nova propriedade que antes só era possível com recurso a imagens: as sombras. É possível criá-la através da introdução de uma linha de código: `box-shadow`. Esta propriedade permite especificar a distância da sombra na caixa na horizontal e na vertical, o raio de desfoque (*blur*), a distância pela qual ela se difunde (*spread*) e a cor da própria sombra (*color*).

```
div { box-shadow: horizontal vertical blur spread color; }
```

Apesar de pertencer ao módulo *Backgrounds and Borders*, candidato a recomendação desde 2014 (W3C, 2014b), esta propriedade não dispensa a utilização de prefixos. A maioria

dos *browsers* mais recentes já a reconhece, porém, por precaução e para navegadores mais antigos, é aconselhada a sua utilização.

Após a criação destes elementos e a construção das mais diversas páginas, foram desenvolvidas várias páginas da área privada, aquelas a que apenas os utilizadores registados têm acesso, tais como as apresentadas nas Figuras 13 e 14. Estas *views* foram executadas em cooperação direta com os programadores do *back-end*, sendo produzidas em resposta ao que ia fazendo falta nesta parte interna da plataforma.

The image shows a web interface for adding a property listing. On the left, there is a sidebar menu with options: 'Edit profile', 'Photos, symbol and video', 'Trust and check' (highlighted), 'Comments', and 'References'. The main content area is titled 'Personal details' and contains a form with the following fields:

- Property Name (text input)
- Description (text area)
- Link/Groom (text input)
- Address1 (text input)
- Address2 (text input)
- Town (text input)
- State (text input)
- Country (text input)
- Postal Code (text input)
- Price (text input)
- Bail (text input)
- House Type (dropdown menu, currently set to 'Apartment')
- Room Type (dropdown menu)
- Persons (dropdown menu)
- Bed Rooms (dropdown menu)
- Beds (dropdown menu)
- Bath Rooms (dropdown menu)
- Status (dropdown menu)
- Allow Animals (dropdown menu)
- Allow Smoking (dropdown menu)

Below the form, there are two sections:

- Add facilities:** A row of five toggle buttons: Wi-Fi, TV, Kitchen, Air Conditioner, and Garage.
- Add Photos:** A section with a 'Photo' label and a file selection button labeled 'Escolher ficheiro' with the text 'Nenhum ficheiro selecionado' below it.

Figura 13: Formulário para a introdução de um anúncio na área interna da plataforma

My Properties


Currently Reserved

Add Property

Statistics

Property Details

Property - Casarão!



Property Details

Property:	Casarão	State:	Santarém
House Type:	Apartment	Address:	Rua das Orquídeas
Room Type:	Room	Other address:	none
Description:	Bonitinho	Postal Code:	2300-575
City:	Tomar		

House Specific Details

Bedrooms:	1	Persons:	2
Single Beds:	4	Bathrooms:	1
Double Beds:	0		

Facilities

<ul style="list-style-type: none"> 📶 Wi-Fi 🍳 Kitchen 	<ul style="list-style-type: none"> 📺 TV
--	---

Location

Figura 14: Detalhes de uma propriedade anunciada no Aweshome, visível apenas para utilizadores registados

5.3 Considerações Finais

Este projeto foi importante na percepção de trabalho em grupo na medida em que, depois de desenhados os *layouts*, era importante que existisse uma cooperação forte entre o *front* e o *back-end*. A gestão de tempos e a junção de trabalhos foi essencial para que o desenvolvimento da plataforma fosse bem-sucedido.

Muitas das vezes, devido a erros de estruturação ou falhas na usabilidade, foi necessária a reformulação e reestruturação de várias páginas. Isto levou a que o projeto se arrastasse durante muito tempo e exigisse muito esforço de todo o grupo.

Foi talvez o projeto que requereu mais atenção e mais experiência transmitiu na área dos *websites*. Demonstrou também a importância que a opinião de um cliente tem no decorrer do projeto. Por vezes, em ambientes académicos, mantemos a nossa ideia, mesmo que isso custe menos um ou dois valores na nota final. Profissionalmente, é necessário manter o cliente satisfeito e desenvolver o projeto segundo as suas diretrizes, por mais que pareçam rígidas ou absurdas.

6 | Motion Graphics

Este trabalho foi um desafio proposto pela empresa de forma a expandir o seu negócio: a realização de um vídeo que seria explicativo e ao mesmo tempo promocional. Ou seja, seria um vídeo que traduziria a importância dos *motions graphics* e a sua ajuda na extensão do negócio. Paralelamente, mostrava que a Sketchpixel os desenvolvia, podendo daí surgir novos trabalhos.

6.1 Tecnologias Utilizadas

Neste projeto foram utilizados os softwares *Adobe Illustrator* e *Adobe After Effects*. O primeiro foi utilizado para definir uma imagem gráfica e para desenvolver algumas formas e figuras que foram posteriormente exportadas e utilizadas no programa de edição de vídeo.

O *Adobe After Effects* foi o *software* utilizado para desenvolvimento da animação. Esta opção deve-se sobretudo à sua composição em *layers*, tal como o *Adobe Photoshop* e o *Adobe Illustrator*. Esta estrutura, para além de facilitar quando se está familiarizado com outros programas da *Adobe*, permite também que cada objeto seja tratado individualmente, podendo ser-lhe aplicado um extenso conjunto de efeitos.

Ainda a sua possibilidade de criar formas e tipografia em vetores foi uma vantagem na sua utilização. Desta forma, muitos dos objetos não necessitaram de exportação do *Adobe Illustrator* e foram diretamente criados no *After Effects*.

6.2 Processo de Desenvolvimento

Este projeto passou por várias fases e contou com a participação da *designer* da empresa. A primeira parte foi a da pesquisa, ver o que se faz nesta área e definir as frases a incluir no vídeo. Para isso foram vistos outras animações do mesmo estilo e alguns conceitos teóricos a elas associadas. Foram então definidas dezassete frases que serviram de guião para o filme:

1. Já pensou em usar vídeos para promover o seu produto?
2. As grandes empresas como *Google*, *Microsoft*, *Apple*, *Yahoo!* utilizam vídeos dinâmi-

cos nas suas campanhas.

3. Com um vídeo o seu cliente retém 58% de mais informação e a hipótese de ganhar mais um cliente é de 6x mais.
4. Visitantes são 85% mais persuadidos a comprar depois de assistir a um vídeo do produto.
5. Vídeos conseguem condensar informação complexa em algo visualmente cativante, acessível e envolvente.
6. Fáceis de ver, compreender e partilhar.
7. Estudos mostram que 62% dos resultados das pesquisas do *Google* apresentam vídeos.
8. *Motion graphics* são simplesmente bons demais para não aproveitar.
9. Vídeos tornam anúncios mais dinâmicos do que apenas informação estática e aborrecida.
10. Vídeos permitem ampliar o seu negócio em 60%.
11. Conteúdo com vídeo atrai 3x mais visitantes.
12. Você pode facilmente atrair seu público-alvo e deixá-los aproveitar o que a sua empresa oferece.
13. Sabe que, se a sua empresa ainda não tem vídeos, está atrás dos seus concorrentes?
14. O utilizador permanece mais tempo num site com vídeo e aumenta o seu lucro.
15. Vídeos juntam interação e envolvimento num curto espaço de tempo.
16. É hora de ver o que os *motion graphics* podem fazer pelo seu negócio:
17. Deixe isso com a *Sketchpixel!*

Após aprovação deste guião, foi-nos pedido um *storyboard* com desenhos que traduzissem visualmente aquilo que foi pensado para cada frase. Esse *storyboard* é apresentado na Figura 15.

Depois disto, foram desenhadas três imagens que iriam futuramente fazer parte do vídeo. Esta fase foi essencial na conceção e desenvolvimento da imagem gráfica (Figura 16). A partir daqui, todos os elementos tiveram de seguir a linha criada. Ou seja, toda a animação teria de seguir o mesmo estilo definido nestes primeiro ecrãs.

Ao dar início à composição do vídeo, a primeira decisão foi de não integrar de frases no vídeo e concebê-lo apenas com imagens. No entanto, certas partes não eram perceptíveis e a animação teve de ser recomeçada, integrando desta vez as frases descritas acima.

Nesta fase, todos os elementos foram animados com base em *keyframes* (Figura 17), sem

serem incluídas animações automáticas incorporadas em bibliotecas do *Adobe After Effects*. Esta decisão foi tomada por se considerar que o vídeo ficaria mais fluido e dinâmico.

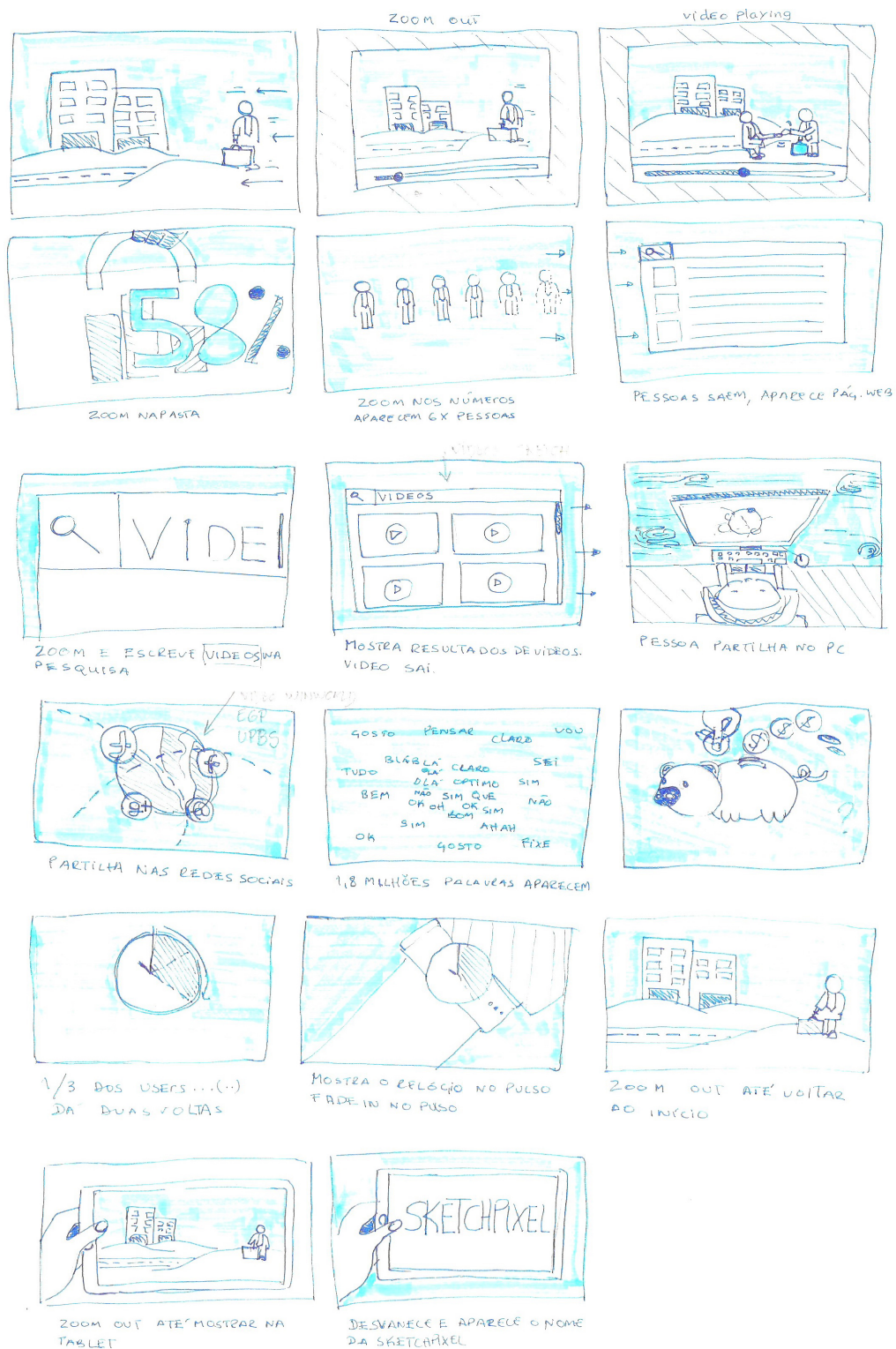


Figura 15: Storyboard para o vídeo motion graphics



Figura 16: Ecrãs para definição da imagem gráfica

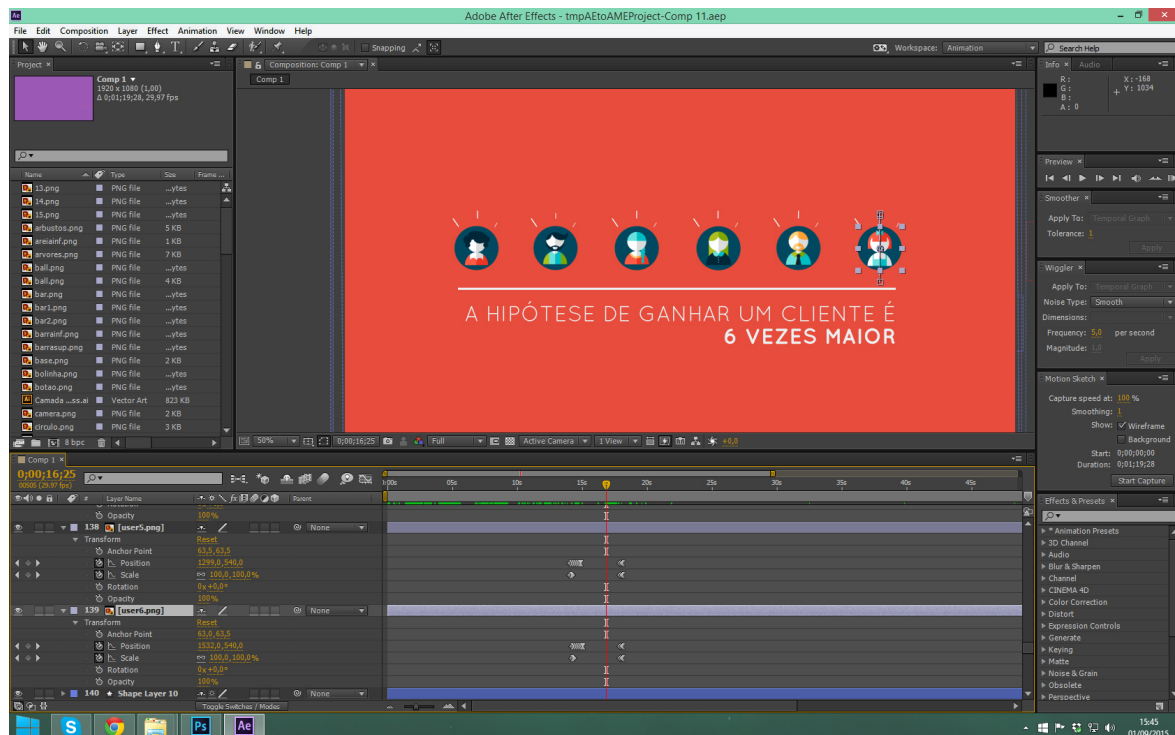


Figura 17: Exemplo exemplificativo da animação em *keyframes* no *Adobe After Effects*

6.3 Considerações Finais

Este foi um trabalho diferente de todos os outros, com outros objetivos e técnicas. Foi uma experiência enriquecedora, sobretudo pela exploração de *softwares* que não foram utilizados antes. Para além disso, seria o início do exercício de uma nova atividade por parte da *Sketchpixel*.

Também a possibilidade de criação de todo o conceito do projeto, a pesquisa e a sua estruturação foi uma nova experiência visto que em todos os outros projetos eram de início fornecidos os objetivos. Assim, foi possível uma maior liberdade e criatividade no processo de desenvolvimento.

Este vídeo foi terminado mas ainda são necessários alguns ajustes. Para além disso, não chegou a um consenso final devido a outros projetos da empresa que se sobrepuseram, nomeadamente o *Aweshome* (explicado no capítulo anterior).

7 | **WORLD OF DISCOVERIES - 'CEUTA ONTEM. CEUTA HOJE.'**

Este projeto consistiu na criação de conteúdos para a exposição 'CEUTA ONTEM. CEUTA HOJE.' do museu *World of Discoveries*, no Porto. Esta é uma exposição comemorativa dos 600 anos da conquista de Ceuta pelos portugueses, inaugurada a 10 de Junho e com fim a 29 de Maio de 2016.

Contou com a colaboração de profissionais das mais diversas áreas. À *Sketchpixel* coube a produção de todos os conteúdos digitais e gráficos, que passou por catálogos, ecrãs *touch*, pirâmides holográficas, vinis, etc.

Este trabalho foi talvez o que mais envolvimento e dedicação exigiu. Foi realizado em Braga, na sede da *Sketchpixel* e permitiu também conhecer alguns colegas com quem era realizado trabalho à distância.

A exposição foi organizada numa cooperação entre a Cidade Autónoma de Ceuta e o Arquivo Nacional da Torre do Tombo, com o intuito de evocar aquele que, em certa medida, pode ser considerado o primeiro passo na História da Expansão Portuguesa. Esta cidade que até 1668 foi portuguesa e mantém ainda, hoje em dia, fortes laços de proximidade com Portugal.

7.1 Livro Digital: *Crónica da Tomada de Ceuta*

O primeiro projeto realizado para esta exposição foi um ecrã *touch* cujo conteúdo era um livro digital. Este correspondia à *Crónica da Tomada de Ceuta* escrito por Gomes Eanes de Zurara. Foi necessária a sua criação visto que o livro em questão faz parte do Arquivo Nacional da Torre do Tombo. O original está também presente na sala de exposição mas permanece intocável devido ao seu valor. Através do ecrã, é dada a possibilidade aos visitantes de conhecerem o livro, manuseá-lo e analisar o seu conteúdo.

Através deste *touchscreen*, o livro impresso dispõe de um aliado sem perder o seu local de destaque. Mais do que questionar o fim da impressão de livros, esta zona da exposição explora a possibilidade de produzir um livro através de um meio digital que revolucionou os meios de comunicação. Para além do despertar dos sentidos, este ecrã é a interação com

História, o processo de chegar a uma coisa através de outra.

Começa a ficar evidente que “toque” e “contacto” não se referem apenas à pele, mas ao jogo recíproco dos sentidos: “manter contacto” ou “estabelecer contacto” é algo que resulta do encontro frutífero dos sentidos (McLuan, 2014, p.81-82).

É assim inegável a importância que estas técnicas de comunicação têm para a preservação do património e ao mesmo tempo de transmissão de conhecimento. Com esta plataforma, é possível chegar a um maior número de leitores de uma forma inovadora e com a qual se identifiquem.

Devido ao grande número de páginas que compõem o original da *Crónica da Tomada de Ceuta*, foram escolhidas apenas aquelas com passagens mais importantes. Esta seleção foi realizada também com base em informações que se pretendiam transmitir: foi de início solicitado que houvesse a possibilidade de inserir um botão ao lado dos excertos mais relevantes. Desta forma, podia proceder-se à sua tradução e tornar o conteúdo acessível a todo o público.

Esta foi uma das razões que levou à utilização de uma *framework* JavaScript como base para a tecnologia utilizada. Antes desta decisão, o trabalho foi iniciado com recurso a um *software* para a criação destes trabalhos: *Flip Builder* (<http://pt.flipbuilder.com>). Acontece que introduzir um botão personalizado que desse acesso a conteúdo exterior ao livro não era possível através deste programa. Para além disso, os *layouts* que apresentava limitavam a criatividade e impunham alguns limites.

Desta forma, o projeto foi realizado com base na *framework* *turn.js*, disponível *online* (<http://www.turnjs.com/>), com a respetiva documentação para a sua utilização. Quando associada ao HTML5 e à CSS3 concede um sem número de opções, possibilitando assim, entre outras coisas, a introdução de um fundo, botões e informações e animações personalizadas.

Estas tecnologias permitiram, para além do efeito de livro digital, um trabalho mais livre e mais de acordo com o pretendido para a exposição. A introdução de excertos traduzidos em seis idiomas foi possível através da *tag* `<i>frame</i>`, já disponível em versões anteriores do HTML. Esta *tag* permite a inserção de um ficheiro HTML dentro de outro (W3C, 2014d), tornando possível a mudança apenas do conteúdo, sem qualquer alteração na formatação da

página de estilos. Assim, os diferentes idiomas iam alterando consoante a seleção do visitante (Figura 18).



Figura 18: *Layout* de apresentação dos excertos traduzidos

O botão no qual o visitante pode aceder a estas informações exigiu também alguma atenção. Foi pedido pelos clientes que consistisse numa pena animada próxima do local correspondente à tradução. Após algumas tentativas de GIFs animados, tanto através de *Adobe Flash* como de *Adobe Photoshop*, foi definido que a melhor opção seria a animação através de JavaScript. Esta decisão está diretamente relacionada com o resultado gráfico obtido. A exportação de um GIF com o fundo transparente em *Adobe Flash* não era possível e o resultado final em *Adobe Photoshop* não era uma animação fluente, sendo perceptível a movimentação dos pixels.

Desta forma, foi exportada uma sequência PNG através do *Adobe Flash* que foi inteiramente colocada no documento HTML e animada através de uma função JavaScript. Essa função define a velocidade da animação e as suas repetições. Inicialmente foi definida com uma maior rapidez, porém existiam imagens da sequência que não surgiam dado que o *browser* não tinha tempo de as carregar. Assim, teve de ser definido um intervalo maior entre o aparecimento de cada imagem (Figura 19).



Figura 19: Pena animada junto ao excerto a ser traduzido (lado direito)

Foi ainda introduzido conteúdo no início da plataforma com informação acerca da obra, da sua importância e do seu autor. Estas informações, traduzidas mais uma vez em seis idiomas, são um *link* possível através da tag <a> com o título do livro a bandeira correspondente a cada língua (Figura 20). Para que este conteúdo não se sobrepusesse ao folhear o livro, foi necessário mais uma vez recorrer a JavaScript. Foi criada uma função que define que quando a primeira página é virada, todo o conteúdo da *div* é ocultado. Foi utilizado para isso o método `fadeOut()` que altera progressivamente a opacidade de forma a que o seu desaparecimento fosse natural e fluido.



Figura 20: *Layout* inicial de apresentação do *touchscreen*

Outra das especificidades deste projeto foi a introdução de um tipo de letra que não o do sistema através da propriedade `@font-face`. Existia uma fonte fornecida pelos clientes que era comum em todos os conteúdos. Com este método, foi possível que este ecrã táctil ficasse de acordo os restantes produtos.

Esta propriedade, para além de indicar a família da fonte, fornece ao *browser* o local dos arquivos a utilizar. É ainda necessário que sejam especificados vários formatos da mesma fonte para que haja compatibilidade entre estas e os mais diversos navegadores.

Uma das ferramentas que auxilia a utilização de fontes web é o *website Font Squirrel* (<http://www.fontsquirrel.com/tools/webfont-generator>). Após o *upload* da fonte desejada, é possível fazer o *download* de uma pasta com todos os formatos necessários para a web e com uma folha de estilos que contém todas as regras necessárias à utilização daqueles ficheiros. Desta forma, é possível que não existam problemas de compatibilidade das fontes. A sintaxe básica da propriedade `@font-face` fica no final da seguinte forma:

```
@font-face {
  font-family: 'MyWebFont';
  src: url('webfont.eot');
  src: url('webfont.eot?#iefix') format('embedded-opentype'),
       url('webfont.woff2') format('woff2'),
       url('webfont.woff') format('woff'),
       url('webfont.ttf') format('truetype'),
       url('webfont.svg#svgFontName') format('svg');
}
```

Foi desta forma possível responder a todos os requisitos necessários ao projeto. Através de toda a explicação anterior, é claramente perceptível que o HTML5, CSS3 e JavaScript foram as tecnologias mais indicadas para todas as especificidades do *touchscreen*.

Foi posteriormente criada uma aplicação *offline* por outro elemento da equipa de modo a permitir o acesso aos dados sem ligação à rede. Esta aplicação torna-se funcional com a junção de um arquivo de manifesto com a lista de todos os recursos da plataforma. Assim, sempre que a aplicação é iniciada por parte do Museu, o *browser* irá ler o manifesto, fazer *download* dos recursos e realizar a cache deles localmente.

7.2 Vinil para Pirâmide Holográfica

Ainda na sequência da exposição 'CEUTA ONTEM. CEUTA HOJE', foi tratada digitalmente uma imagem para ser impressa em vinil de forma a cobrir a parte de baixo de uma pirâmide holográfica. Esta pirâmide tinha como objetivo simular um altar presente na Catedral de Ceuta. Esta Catedral era em 1415, ano da Conquista de Ceuta, uma mesquita, porém foi transformada em templo cristão após a conquista por parte dos portugueses.

Assim sendo, a pirâmide holográfica continha uma animação holográfica da Santa em três dimensões e a sua base foi coberta com uma imagem que representava a própria base da Santa presente na Catedral. Esta foi dividida em três partes de forma a cobrir as laterais e a frente da pirâmide.

O tratamento desta imagem foi realizado em *Adobe Photoshop* e o principal desafio consistiu no aumento de uma imagem *bitmap*. Como foi explicado no capítulo 3.5, as imagens fotográficas, que por defeito são compostas por pixéis, não devem ser reduzidas nem aumentadas.

Acontece que a imagem fornecida tinha à volta de quarenta centímetros de altura e para cobrir toda a pirâmide eram necessários 1,20 metros.

Após o reenquadramento da imagem, correções e alterações cromáticas, aumento de brilho e contraste, houve a necessidade de recorrer ao filtro *High Pass*. Este filtro permite manter os detalhes onde ocorrem transições nítidas de cores e suprime a restante imagem. Ou seja, remove o detalhe de baixa frequência, dando a ilusão de que a imagem está definida.

A vantagem da utilização deste filtro nesta imagem deve-se ao seu efeito desvanecido aquando da ampliação da imagem. Tal como visto no capítulo 3.5, quando uma imagem *bitmap* é aumentada, são criados novos pixéis com base nos adjacentes, eliminando detalhes da imagem.

Foi por fim dividida a imagem em três partes (para as laterais e frente) com o cuidado de manter uma margem para cada lado de modo a que nenhuma parte ficasse por revestir e que as três partes pudessem juntar-se na perfeição (Figura 21).



Figura 21: Imagens finais para impressão em vinil

7.3 *Layouts*: ‘Ceuta em 1415’ | ‘Ceuta Portuguesa’

Por fim, o último trabalho realizado para a exposição foi o *design* de dois ecrãs *touch* que, apesar de distintos, são complementares. Estes *layouts* foram colocados junto de mapas da cidade de Ceuta: uma maquete de Ceuta antes da ocupação portuguesa (‘Ceuta em 1415’) e um mapa da cidade atual, impresso em vinil e colado numa parede (‘Ceuta Portuguesa’).

O objetivo destes ecrãs é corresponder a cada um dos mapas através de números, explicando o local (desde monumentos, passando por museus e objetos), a sua importância e valor. A cada número correspondia então um espaço que continha imagens e alguma informação sobre o local em questão.

A opção de desenhar os dois *layouts* idênticos foi sobretudo para manter a coerência entre os espaços da exposição, não obrigando o visitante a ter de interpretar duas estruturas diferentes.

Estes ecrãs foram produzidos através do *software Adobe Illustrator* por se considerar mais simples qualquer alteração necessária nas formas utilizadas. Para além disso, o desenho em vetores é mais facilmente redimensionado. Assim, foram primeiramente tratadas e reenquadradas todas as imagens fotográficas em *Adobe Photoshop* e posteriormente importadas para o *Adobe Illustrator*.

Os *layouts* foram desenhados com três momentos distintos: uma página inicial, outra de seleção do ponto desejado e por último as imagens do espaço em *full-screen* com um apontamento textual explicativo.

O ecrã inicial pretende dar a conhecer o espaço em que o visitante se encontra, com o nome do mesmo e a opção de selecionar o idioma pretendido. Esta opção é dada apenas no início para que o visitante navegue por todas as páginas e veja tudo o que o *touchscreen* em questão pretende transmitir. Caso esta opção existisse na página em que se seleciona o número, por exemplo, o visitante não teria acesso ao nome do espaço e poderia não fazer sentido nem se perceber o contexto daquela plataforma.

O segundo momento destes ecrãs tem uma aparência de *timeline* onde se encontram imagens em miniatura dos espaços e por baixo os números correspondentes à maquete ou ao mapa. Aqui pode ver-se também o nome dos monumentos, objetos ou museus e ter-se uma noção geral do que se pretende transmitir.

Por último, a página com imagens em *fullscreen* deixam apreciar mais de perto os detalhes do espaço. Têm a opção de ler um pouco sobre o ponto em questão, transmitindo maior conhecimento ao visitante. Esta informação é facilmente ocultada através de um símbolo com uma cruz para fechar, de forma a que a imagem possa ser novamente vista após a leitura do texto.

O ecrã correspondente a ‘Ceuta em 1415’ tem um tom mais avermelhado devido à imagem fornecida para colocar como fundo. Esta representa um mapa desenhado no século XV com Portugal e todo o Norte de África.

No segundo *layout*, ‘Ceuta Portuguesa’, houve uma maior liberdade de criação, sendo utilizada uma imagem de uma construção original portuguesa, importante para a defesa da cidade (Baluarte dos Maiorquinhos). Foi aplicado um filtro na imagem, dando-lhe uma tonalidade mais térrea para ficar de acordo com as cores da exposição.

Assim, pode afirmar-se que estes dois ecrãs mantêm a forma e diferem apenas no conteúdo. As cores utilizadas são também diferentes para que à primeira vista não pareça ao visitante que a informação contida é a mesma (Figura 22 e Figura 23).



Figura 22: Layout ‘Ceuta em 1415’

Figura 23: Layout ‘Ceuta Portuguesa’

7.4 Considerações Finais

Todos os trabalhos desenvolvidos para a exposição ‘Ceuta Ontem. Ceuta Hoje’ permitiram explorar um conjunto de técnicas, *softwares* e tecnologias não utilizadas anteriormente. A necessidade de terminar todos os conteúdos para a exposição num curto espaço de tempo mostrou que, por vezes, a atenção ao pormenor, o rigor e a persistência são maior auxílio que o conhecimento de muitos meios.

Não houve a oportunidade de visitar a exposição no museu *World of Discoveries*, apenas foram vistos os ecrãs terminados antes da sua montagem. Através de fotografias e vídeos, percebeu-se que a sala ficou bem organizado e que os mais diversos conteúdos se interligavam no sentido de dar a conhecer à comunidade a importância histórica da Conquista de Ceuta (Figura 23).



Figura 24: Sala da exposição 'CEUTA ONTEM. CEUTA HOJE.' no museu *World of Discoveries* no Porto
Fonte: Página da *Sketchpixel* no *Facebook*⁵

⁵Disponível em <URL: <https://www.facebook.com/sketchpixel>>

8 | Conclusão

A explosão da informação e a necessidade da sua rápida disseminação levou a uma alteração na sociedade do fixo para o móvel, transformando a produção, distribuição e gestão de todos os conteúdos digitais.

Ao longo de todo o mestrado foi-nos transmitida a importância da junção de criativos e programadores nos mais diversos projetos de forma a responder às necessidades digitais. Saber trabalhar com vários tipos de pessoas das mais diversas áreas é importante para a criação de conteúdos que vão ao encontro daquilo que o público precisa e anseia.

Considerou-se desde início importante explorar uma grande diversidade de técnicas de forma a desafiar e criticar o trabalho para um melhor desenvolvimento do mesmo. Cada projeto foi como que um exercício ao serviço de uma aprendizagem constante, tanto a nível pessoal como intelectual.

Apesar de nenhum destes projetos ter sido uma proposta totalmente inovadora, visto que todos eles foram solicitados por parte de clientes ou por necessidades da própria *Sketchpixel*, foram aplicados vários conhecimentos. Este era um dos objetivos iniciais e que foi concretizado da melhor forma possível com projetos que, ainda que estabelecidos, trouxeram entusiasmo e desafios.

Particularmente nos dois primeiros projetos descritos neste relatório, *Aweshome* e *XPhoto*, o trabalho desenvolvido foi o elo de ligação entre *designer* e programador. A programação do código *front-end* foi um desafio não só pela razão acima descrita mas também pela necessidade de gerir os tempos de forma a que todo o trabalho fosse bem sucedido e sem grandes demoras.

A realização de um vídeo em *motion graphics*, para além de ser uma experiência totalmente nova, foi o trabalho que mais liberdade de criação proporcionou. No entanto, com todas as suas possibilidades, foi o que mais pesquisa conceptual exigiu. Para além disso, foi necessária uma análise criteriosa de todo o conjunto gráfico e de cada elemento a inserir.

Por último, a exposição ‘CEUTA ONTEM. CEUTA HOJE.’, talvez o trabalho mais ponderado e com mais responsabilidade, exigiu um grande foco dado o tempo para a sua realização. Possibilitou também conhecer a sede da *Sketchpixel* e trabalhar pessoalmente com colegas com os quais ainda só tinha existido a oportunidade de trabalhar à distância através de diversos meios digitais. Também o facto de se tratar de conteúdos para uma exposição aberta ao público, fez deste trabalho único.

Numa última análise ao estágio, é possível dizer que estar numa rotina de trabalho com horários e objetivos para cumprir foi imprescindível para a realização do presente relatório. Também foi importante para perceber a necessidade de encontrar novos métodos e mecanismos de trabalho, sempre com o objetivo de os tornar mais rápidos e eficazes. É indispensável nas empresas como a *Sketchpixel* provar o seu valor no mercado através do rigor, persistência e paciência.

Referências

- BENGTSON, J. (2013) 'What Is HTML5 and What Do You Need To Know About It?' *Journal of Hospital Librarianship*. 13:4, p. 392-398.
- BERNERS-LEE, T. (2006) *Reinventing HTML*. [Consult. 2015-06-26]. Disponível em <URL: <http://dig.csail.mit.edu/breadcrumbs/node/166>>
- BLAGDEN, R. (2014) *Why Motion Graphics are this year's 'must-have'*. [Consult. 2015-04-26]. Disponível em <URL: <http://torpedogroup.com/blog/post/why-motion-graphics-are-this-years-must-have>>
- CROFT, J. (2007) *Frameworks for Designers*. [Consult. 2015-09-26]. Disponível em <URL: <http://alistapart.com/article/frameworksfordesigners>>
- EVENING, M. (2015) *Image essentials*. [Consult. 2015-07-03]. Disponível em <URL: <https://helpx.adobe.com/photoshop/using/image-essentials.html>>
- LAWSON, B., SHARP, R. (2011) *Introducing HTML5*. Berkeley: New Riders.
- LAWSON, B. (2009) *Interview with Ian Hickson, editor of the HTML 5 specification*. [Consult. 2015-09-03]. Disponível em <URL: <http://www.webstandards.org/2009/05/13/interview-with-ian-hickson-editor-of-the-html-5-specification/>>
- MARCOTTE, E. (2010) *Responsive Web Design*. [Consult. 2015-08-25]. Disponível em <URL: <http://alistapart.com/article/responsive-web-design>>
- GASSTON, P. (2011) *The Book of CSS3 – A Developer's Guide to the Future of Web Design*. São Francisco: No Starch Press.
- MCLUAN, M. (2014) *Os Meios de Comunicação como Extensões do Homem*. São Paulo: Editora Cultrix.
- MEYER, E. (2007) *Reset Reasoning* [Consult. 2015-08-03]. Disponível em <URL: <http://meyerweb.com/eric/thoughts/2007/04/18/reset-reasoning/>>
- MIRZOEFF, Nicholas (1999). *An Introduction to Visual Culture*. Londres e Nova York: Routledge.

- MITCHELL, E. (2010) 'Standards, efficiency, and the Evolution of Web Design' *Journal of Web Librarianship*. 4:4, p. 453-455.
- SHAW, P. (2015) *CSS3 Succinctly*. Morrisville: Syncfusion.
- SILVA, M. (2010) *JavaScript – Guia do Programador*. São Paulo: Novatec Editora.
- SILVA, M. (2012) *CSS3 – Desenvolva aplicações web profissionais com o uso dos poderosos recursos de estilização das CSS3*. São Paulo: Novatec Editora.
- SILVA, M. (2014a) *HTML5 – a Linguagem de Marcação que Revolucionou a Web*. São Paulo: Novatec Editora.
- SILVA, M. (2014b) *Web Design Responsivo*. São Paulo: Novatec Editora.
- TUCK, M. (2010) *The History of CSS Resets* [Consult. 2015-08-03]. Disponível em <URL: <http://sixrevisions.com/css/the-history-of-css-resets/>>
- VARGAS, H., Souza, L. (2011) 'A colagem como processo criativo: da arte moderna ao motion graphics nos produtos midiáticos audiovisuais' *Revista Comunicação Midiática*. 6:3, p.51-70.
- W3C (2005) *Document Object Model (DOM)* [Consult. 2015-09-20]. Disponível em <URL: <http://www.w3.org/DOM/>>
- W3C (2012) *Media Queries* [Consult. 2015-08-20]. Disponível em <URL: <http://www.w3.org/TR/css3-mediaqueries/>>
- W3C (2014a) *HTML5* [Consult. 2015-08-30]. Disponível em <URL: <http://www.w3.org/TR/html5/>>
- W3C (2014b) *CSS Backgrounds and Borders Module Level 3* [Consult. 2015-09-15]. Disponível em <URL: <http://www.w3.org/TR/css3-background/>>
- W3C (2014c) *JAVASCRIPT WEB APIS* [Consult. 2015-09-23]. Disponível em <URL: <http://www.w3.org/standards/webdesign/script>>
- W3C (2014d) *The iframe element* [Consult. 2015-09-25]. Disponível em <URL: <http://www.w3.org/TR/html5/embedded-content-0.html#the-iframe-element>>
- YA-MEI, Z., JUN-JIE, C. (2010) 'Research on Method of Transformation from Bitmap to Vector Graphics based on Adobe Illustrator CS4' *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. Volume 3, p.75-77.