



Estudo de Funcionalidades HTML5 para um Editor de Imagem na Internet

AGOSTINHO MANUEL PEREIRA MONTEIRO

Outubro de 2015

Estudo de Funcionalidades HTML5 para um Editor de Imagem na Internet

Agostinho Manuel Pereira Monteiro

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Doutor Filipe Pacheco

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, [Mês] 2015

Para a minha Família

Resumo

Hoje em dia existem múltiplas aplicações multimédia na Internet, sendo comum qualquer website apresentar mais de uma forma de visualização de informação além do texto como, por exemplo: imagens, áudio, vídeo e animação.

Com aumento do consumo e utilização de *Smartphone* e *Tablets*, o volume de tráfego de internet móvel tem vindo a crescer rapidamente, bem como o acesso à internet através da televisão. As aplicações web-based ganham maior relevância devido à maior partilha ou consumo de conteúdos multimédia, com ou sem edição ou manipulação da mesma, através de redes sociais, como o Facebook.

Neste documento é apresentado o estudo de alternativas HTML5 e a implementação duma aplicação web-based no âmbito do Mestrado de Engenharia Informática, ramo de Sistemas Gráficos e Multimédia, no Instituto Superior Engenharia do Porto (ISEP). A aplicação tem como objetivo a edição e manipulação de imagens, tanto em desktop como em dispositivos móveis, sendo este processo exclusivamente feito no lado do cliente, ou seja, no Browser do utilizador. O servidor é usado somente para o armazenamento da aplicação.

Durante o desenvolvimento do projeto foi realizado um estudo de soluções de edição e manipulação de imagem existentes no mercado, com a respetiva análise de comparação e apresentadas tecnologias Web modernas como HTML5, CSS3 e JavaScript, que permitirão desenvolver o protótipo.

Posteriormente, serão apresentadas, detalhadamente, as várias fases do desenvolvimento de um protótipo, desde a análise do sistema, à apresentação do protótipo e indicação das tecnologias utilizadas. Também serão apresentados os resultados dos inquéritos efetuados a um grupo de pessoas que testaram esse protótipo.

Finalmente, descrever-se-á de forma mais exaustiva, a implementação e serão apontadas dificuldades encontradas ao longo do desenvolvimento, bem como indicadas futuras melhorias a introduzir.

Palavras-chave: HTML5, CSS3, Canvas, JavaScript, Edição de imagens, Manipulação de imagem, Aplicações Multimédia, Aplicações Online.

Abstract

Nowadays there are several multimedia applications in the internet, where it is usual on any website to show more than one visual form of information besides text such as audio, video, animation and images.

With the massification of Smartphone and Tables, the volume of internet traffic has been growing fast, as well as the internet access by SmartTVs and similar devices. The web-based applications gain more impact because more multimedia content has been shared, with or without edition or manipulation, on social networks like Facebook.

In this document is presented the implementation of web-based application on context of Computer Engineering Master Degree, Graphics Systems and Multimedia, at Instituto Superior Engenharia do Porto (ISEP). The application aims to do image edition and manipulation, in desktop as well as in mobile devices, where the process is exclusive of client side, which means, on user browser application. The server is used only to storage the resources of the application.

To develop the project a study of edition and manipulation image software in the market was performed, with the due analysis of comparison and the presentation of web technologies as HTML5, CSS3 and JavaScript, that will help to develop the prototype.

After, it will be shown in detail, the phases of the development of the prototype, from system analysis to visual presentation of the prototype and which technologies were chosen. It is also presented the results of the surveys made to a group of people that tested the prototype.

Finally, it will be thoroughly described the implementation and it will be indicated some difficulties during the development, as well as the future evolutions to insert.

Keywords: HTML5, CSS3, Canvas, JavaScript, Image Edition, Image Manipulation, Multimedia Application, Web-Based Application.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao orientador, Professor Doutor Filipe Pacheco, pelos conhecimentos transmitidos e apoio científico prestado, assim como todo o empenho, dedicação, disponibilidade e paciência demonstrados. A sua participação ativa tornou esta investigação uma experiência verdadeiramente enriquecedora a nível académico e profissional.

A todos os alunos do mestrado em Engenharia Informática do ISEP que testaram e avaliaram o protótipo preenchendo os respetivos inquéritos.

À minha família, em especial aos meus pais, pelo apoio, fraternidade e compreensão total nos momentos em que estive mais ausente.

A todos os que referi, e aos que possa não ter referido, o meu mais sincero obrigado.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Estrutura	2
2	Estado da Arte	4
2.1	Software de Edição e Manipulação de Imagem	4
2.1.1	Software Selecionado	4
2.1.2	Análise Comparativa	17
2.1.3	Conclusão	20
2.2	Tecnologias Web	21
2.2.1	Adobe Flash	21
2.2.2	HTML	24
2.2.3	CSS	33
2.2.4	JavaScript	37
2.2.5	Bibliotecas e Frameworks	40
2.2.6	Conclusão	50
3	Projeto	52
3.1	Análise do Sistema	53
3.1.1	Arquitetura	53
3.1.2	Funcionalidades	54
3.1.3	Casos de Uso	56
3.1.4	MockUps	58
3.1.5	Website oficial do projeto	61
3.2	Protótipo	63
3.2.1	Estrutura de Navegação	63
3.2.2	Tecnologias e Software	64
3.2.3	Plataformas	66
3.2.4	Cronograma	66
3.2.5	Módulos	68
3.2.6	Interface com Utilizador	71
3.2.7	Avaliação	74
3.3	Implementação	90
3.3.1	Módulo GUI	90
3.3.2	Módulo Ficheiros	93
3.3.3	Módulo Essenciais	97
3.3.4	Módulo Ajustamentos	101
3.3.5	Módulo Efeitos	104
3.3.6	Módulo Texto	106
3.3.7	Módulo Ecrã inteiro	108
3.4	Conclusão	110
4	Conclusão	113

4.1	Resumo e objetivos alcançados	113
4.2	Trabalho Futuro.....	114

Lista de Figuras

Figura 1 - Adobe Photoshop: Workspaces	5
Figura 2 - Adobe Photoshop: Painel da Esquerda (Ferramentas)	5
Figura 3 - Adobe Photoshop: Painel da Direita	6
Figura 4 - Adobe Photoshop CC, versão Mac OS X.....	7
Figura 5 - Interface GIMP, versão Mac OS X.....	9
Figura 6 - Interface Paint.Net	10
Figura 7 - Interface Sumo Paint.....	11
Figura 8 - Interface Pixlr, versão Mac OS X	13
Figura 9 - Interface Fotor, versão Mac OS X.....	14
Figura 10 - Interface Picozu	16
Figura 11 - A quick tour of Flash's Genealogy Tree [Yassine Bouhiel, 2010].....	22
Figura 12 - Cronologia das tenologia HTML e CSS.....	25
Figura 13 - SVG: Demonstração.....	28
Figura 14 - HTML5 Canvas: Código Exemplo	29
Figura 15 - Imagem Raster / bitmap.....	31
Figura 16 - Imagem vectorial.....	31
Figura 17 - CSS: Aplicar CSS	34
Figura 18 - JavaScript: Incorporar.....	38
Figura 19 - jQuery: Exemplos	40
Figura 20- CamanJS: Demonstração.....	42
Figura 21 - FabricJS: Demonstração	43
Figura 22 - EaselJS: Demonstração.....	44
Figura 23 - HammerJS: Demonstração	45
Figura 24 - Raphael: Demonstração	46
Figura 25 - Snap SVG: Demonstração.....	47
Figura 26 - Zurb Foundation: Código Exemplo	48
Figura 27 - Twitter Bootstrap: Exemplo Código	49
Figura 28 - Arquitetura do Sistema Proposto.....	53
Figura 29 - Casos de Uso do sistema	56
Figura 30 - MockUp de Organização	58
Figura 31 - Mockup do Layout da Aplicação	59
Figura 32 - MockUp do Layout (Sub-Menu)	60
Figura 33 - Website Oficial do Projecto.....	61
Figura 34 - Estrutura de Navegação da Protótipo.....	63
Figura 35 - Cronograma de Desenvolvimento.....	66
Figura 36 - User Interface: Desktop.....	71
Figura 37 - User Interface: Dispositivos Móveis.....	72
Figura 38 - User Interface: Dispositivos Móveis (Menu)	73
Figura 39 - Website direcionado a testers	74
Figura 40 - Inquérito 1: Geral	76

Figura 41 - Inquérito 1: Desktop	77
Figura 42 - Inquérito 1: Sistemas Móveis.....	78
Figura 43 - Inquérito 1: Software Online.....	79
Figura 44 - Inquérito 2: Geral	81
Figura 45 - Inquérito 2 – Teste em Desktop.....	82
Figura 46 - Inquérito 2: Teste Visual em Desktop	83
Figura 47 - Inquérito 2: Funcionalidades Implementadas em Desktop	83
Figura 48 - Inquérito 2: Funcionalidades a Implementar em Desktop	84
Figura 49 - Inquérito 2: Teste em dispositivos móveis.....	85
Figura 50 - Inquérito 2: Tracker Google Analytics.....	85
Figura 51 - Inquérito 2: Teste visual em Dispositivos Móveis	86
Figura 52 - Inquérito 2: Funcionalidades em dispositivos Móveis.....	87
Figura 53 - Inquérito 2: Uso futuro	88
Figura 54 - Módulo Menu: Inicial	90
Figura 55 - Módulo Menu: Essenciais	91
Figura 56 - Módulo GUI: Código do Menu	91
Figura 57 - Módulo GUI: Código Adaptação das dimensões do Canvas	92
Figura 58 - Módulo Ficheiros: Menu Image	93
Figura 59 - Módulo Imagem: Código Upload Imagem do Disco	94
Figura 60 - Módulo Imagem: Código Captação de Imagem Webcam	95
Figura 61 - Módulo Essenciais: Crop	97
Figura 62 - Módulo Essenciais: Seleção Crop.....	98
Figura 63 - Módulo Essenciais: Zoom e Pan.....	99
Figura 64 - Módulo Essenciais: Erro em Zoom.....	100
Figura 65 - Módulo Ajustamentos: Menu	101
Figura 66 - Módulo Ajustamentos: Código	102
Figura 67 - Módulo Efeitos: Código.....	104
Figura 68 - Módulo Efeitos: Menu	105
Figura 69 - Módulo Texto: Propriedade do Texto	106
Figura 70 - Módulo Texto: código do Core.....	107
Figura 71 - Módulo Fullscreen: Código	109
Figura 72 - Anexo 1: Formulário 1 (Pág. 1).....	121
Figura 73 - Anexo 1: Formulário 1 (Pág. 2).....	122
Figura 74 - Anexo 1: Formulário 1 (Pág. 3).....	123
Figura 75 - Anexo 1: Formulário 1 (Pág. 4).....	124
Figura 76 - Anexo 1: Formulário 1 (Pág. 5).....	125
Figura 77 - Anexo 1: Formulário 1 (Pág. 6).....	126
Figura 78 - Anexo 1: Formulário 1 (Pág. 7).....	127
Figura 79 - Anexo 1: Formulário 1 (Pág. 8).....	128
Figura 80 - Anexo 2: Formulário 2 (Pág. 1).....	129
Figura 81 - Anexo 2: Formulário 2 (Pág. 2).....	130
Figura 82 - Anexo 2: Formulário 2 (Pág. 3).....	131
Figura 83 - Anexo 2: Formulário 2 (Pág. 4).....	132

Figura 84 - Anexo 2: Formulário 2 (Pág. 5).....	133
Figura 85 - Anexo 2: Formulário 2 (Pág. 6).....	134
Figura 86 - Anexo 2: Formulário 2 (Pág. 7).....	135
Figura 87 - Anexo 2: Formulário 2 (Pág. 8).....	136

Lista de Tabelas

Tabela 1 - Forma de Importação e Exportação	17
Tabela 2 - Formato de Importação e Exportação.....	18
Tabela 3 - Funcionalidades	18
Tabela 4 - Partilha.....	19
Tabela 5 - Plataformas.....	19
Tabela 6 - Diferenças entre sistemas gráficos.....	32

Acrónimos e Símbolos

Lista de Acrónimos

API	<i>Application Programming Interface</i>
CAD	<i>Computer- Aided Design</i>
CaManJS	<i>Canvas Manipulation in JavaScript</i>
CC	<i>Creative Cloud</i>
CORS	<i>Cross-Origin Resource Sharing</i>
CSS	<i>Cascading Style Sheet</i>
DOM	<i>Document Object Model</i>
ECMA	<i>European Computer Manufacturers Association</i>
GIMP	<i>GNU Image Manipulation Program</i>
GNU	Acrónimo recursivo: “GNU’s Not Unix”
GUI	<i>Graphic User Interface</i>
HTAccess	<i>Hypertext access</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
SEO	<i>Search Engine Optimization</i>
SMS	<i>Short Message Service</i>
SGML	<i>Standard Generalized Markup Language</i>
SVG	<i>Scaling Vector Graphics</i>
URL	<i>Uniform Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
WHATWG	<i>Web Hypertext Application Technology Group</i>
XML	<i>eXtensible Markup Language</i>

1 Introdução

1.1 Enquadramento

No século passado a distribuição de informação multimédia era muito limitada, com sistemas de informação baseados em texto. No entanto, os utilizadores queriam mais para além de informação impressa. Graças aos avanços das tecnologias multimédia, tornou-se possível, nos finais do século passado, aceder a qualquer tipo de informação como imagens, vídeo e som [Chen, 2014].

Hoje em dia existem múltiplas aplicações multimédia na Internet, e é comum qualquer website apresentar mais de uma forma de visualização de informação além do texto, tais como imagens, áudio, vídeo e animação [wikibooks, 2015].

A quantidade de visualização de imagens e vídeos através da internet está a crescer a ritmo impressionante, em parte impulsionada pela partilha dessas mesmas imagens e vídeos através de redes sociais, como Facebook e motores de pesquisa como Google, Microsoft Bing e Youtube [Krishna et al., 2010]. No Instagram, os utilizadores já partilharam mais 30 mil milhões de imagens, até outubro de 2014 [Intagram, 2015].

O mesmo acontece na internet móvel com Smartphones e Tablets cada vez mais usados para aceder à internet a nível mundial [Qin Gang, 2014]. De acordo com Bareme Internet da Masktest, 35% dos portugueses acediam à internet através de dispositivos móveis em 2013. Para além destes, há acesso à internet através de televisores interativos [Jorna de Negócios, 2013].

Mas as limitações, nível de *hardware*, dos dispositivos móveis tem causado constrangimentos no que toca a software de processamento de imagem mais avançados [Liang et al, 2009].

Com o abandono dos dispositivos móveis da marca Apple e posteriormente Adobe, da tecnologia Flash [iLex, 2011], as aplicações web baseadas em HTML5 ganharam mais alento, ao ajudar a resolver alguns dos problemas, mais concretamente o problema de compatibilidade nos mais variados tipos de dispositivos [Myeongjin Cho et al, 2014].

Neste documento é apresentada a implementação duma aplicação web-based no âmbito do Mestrado de Engenharia Informática, ramo de Sistemas Gráficos e Multimédia, no Instituto Superior Engenharia do Porto (ISEP). A aplicação tem como objetivo a edição e manipulação de imagens, tanto em desktop como dispositivos móveis, sendo este processo exclusivamente feito no lado do cliente, ou seja, no Browser do utilizador. O servidor é usado somente para o armazenamento da aplicação.

O desenvolvimento do projeto é baseado no estudo de soluções existentes atualmente tanto para desktop, como para dispositivos móveis e aplicações web-based. Além deste estudo foram também realizados testes com um número limitado de utilizadores do protótipo inicial no sentido de orientar o restante desenvolvimento.

Este trabalho permitiu retirar diversas conclusões sobre as potencialidades e limitações das tecnologias web neste contexto.

1.2 Estrutura

Nesta secção é descrita a organização do documento, apresentando de forma sucinta os 4 grandes capítulos.

- **Capítulo 1:** Introdução. Contextualização tanto do documento como do projeto proposto e implementado. Apresentação geral da organização do documento.
- **Capítulo 2:** Estado da Arte. Descrição detalhada de alguns programas de edição e manipulação. Análise comparativa dos programas selecionados. Apresentações de tecnologias Web e respetivas Frameworks ou bibliotecas capazes de implementar certas funcionalidades existentes nos Softwares de edição e manipulação descritos.
- **Capítulo 3:** Projeto. Descrição exhaustiva da aplicação proposta, desde análise em termos estruturais, à apresentação em linhas gerais do protótipo desenvolvido. E análise mais detalhada do seu desenvolvimento.
- **Capítulo 4:** Conclusões. Nova contextualização do projeto. Apresentação das ilações retiradas, do que foi implementado no projeto. Indicação das futuras melhorias que contribuam para a evolução do projeto.

2 Estado da Arte

Neste primeiro capítulo são apresentadas em primeiro lugar algumas aplicações utilizadas atualmente para edição e manipulação de imagem. Em seguida, uma análise comparativa das mesmas. Posteriormente são descritas diversas tecnologias web, disponíveis para o desenvolvimento de aplicações online.

2.1 Software de Edição e Manipulação de Imagem

Ao longo deste capítulo são apresentadas e descritas diversas aplicações. Foi feita uma pesquisa por aplicações para desktop, dispositivos móveis ou via web que se enquadrem no perfil do projeto. Com características que vão desde o tipo de plataforma, tecnologia utilizada e preço. Na descrição de cada aplicação são realizadas características gerais e outras que se destacam das restantes.

2.1.1 Software Selecionado

Adobe Photoshop (A)

Adobe Photoshop, é um dos editores de imagem mais conhecidos e um dos Softwares mais capazes no que toca a edição de imagem [Rod Lawton, 2014]. A sua primeira grande versão foi lançada em 1990 e atualmente vai na versão 14.2, com nome Adobe Photoshop CC (Creative Cloud) [Angela West, 2010].

Algumas novas funcionalidades desta versão incluem as camadas com objetos inteligentes (smart objects), que tem como um dos principais objetivos ligar um imagem a múltiplos projetos que estejam em formato PS. Quando essa imagem é alterada, é atualizada nos diversos projetos [Michael Muchmore, 2014a]

Relativamente à interface gráfica, para além de ter menus mais tradicionais, que aparecem no topo da aplicação, desde File (ficheiro) a *Help* (Ajuda), existem os chamados *Workspace* (ambiente / área de trabalho), com os seguintes nomes predefinidos: *Essentials* (Essenciais), 3D, Motion (Animação), Painting (Pintura), *Photography* (Fotografia), *Typography* (Tipografia). Cada um destes *Workspace* é constituído por um conjunto específico de *Panels* (painéis).

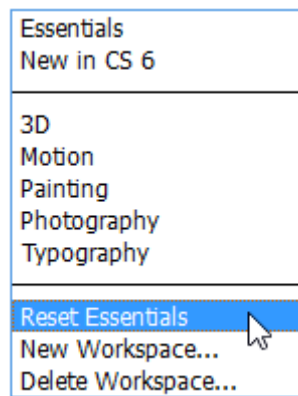


Figura 1 - Adobe Photoshop: Workspaces

Por exemplo, utilizando como referência, o *Workspace Essentials*, este, à semelhança dos restantes, é constituído por 2 painéis principais, um alinhado mais à esquerda (Figura 2), onde se podem encontrar as mais variadas ferramentas, desde seleção à alteração de cor.



Figura 2 - Adobe Photoshop: Painel da Esquerda (Ferramentas)

O segundo painel, figura 3, é constituído por funcionalidades como *Layers* (camadas), texto, paletes de cores, etc. Tanto o painel mais esquerda como naquele mais à direita, é possível alterar o seu conteúdo, adicionar ou remover ferramentas, no caso do painel à esquerda, adicionar e remover sub-painéis no caso do painel da direita.

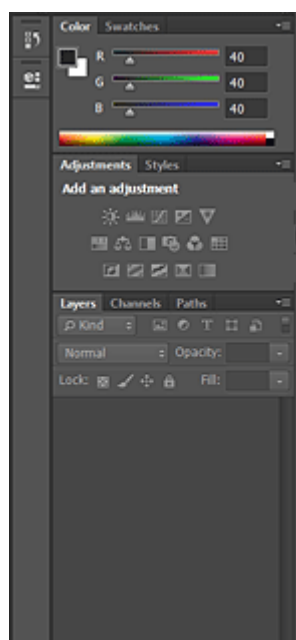


Figura 3 - Adobe Photoshop: Painel da Direita

Existe também um terceiro painel, no topo da imagem, cujo seu conteúdo está dependente da ferramenta selecionada, no painel da esquerda. Este terceiro painel encontra-se abaixo do menu tradicional [Steve Patterson, 2015].

Para além de ferramentas e filtros padrão e possibilidade de filtros 3D, pinturas 3D, há a capacidade de criar uma cena de impressão 3D e, posteriormente, imprimir o modelo caso o utilizador tenha uma impressora com capacidade de impressão 3D.

Uma das novas funcionalidades, diz respeito a utilização de fontes, com possibilidade de pesquisar por uma certa tipo de fonte por letra ou conjunto de letras, bem como se for pressionada tecla *shift*, encontrar família dessa fonte. Além disso, o novo serviço de fontes da Adobe, Adobe Type [Michael Muchmore, 2014b], permite adicionar novas fontes às existentes, no dispositivo em que o utilizador estiver a trabalhar. Caso o utilizador queira visualizar apenas fontes descarregadas do *Adobe Type*, basta selecionar no respetivo ícone, para filtrar pelas fontes de serviço, este serviço contém tanto fontes para desktop como para web.

Outra situação muito comum no desenvolvimento web, é após a conclusão do *Layout* de um Website por parte do Web Designer, ao ser transferido para o *Web Developer*, não incluir respetivas fontes.

A nova versão Photoshop, verifica as fontes existentes no Typekit, questionando o utilizador se deseja descarregar essa(s) fonte(s) em falta. Caso descarregue, ficarão sempre disponível para o utilizador.

Os preços deste serviço podem ir desde 24,99 dólares por ano no plano “*Personal*”, 49,99 dólares por ano para o plano “*Portfolio*”, 99,99 dólares por ano no plano “*Performance*”, por último plano “*Business*” pode ir desde 100 dólares até 400 dólares por ano [TypeKit, 2014].

Outra das novidades, que interessará *Web Developers*, é a possibilidade de após conclusão de um *Layout*, seleccionar uma determinada área do *Layout* e respetivas camadas e, com o rato direito seleccionar a opção “copiar CSS”. Ao colar num editor web, o utilizador verificará, que foi criada uma nova classe, com as respetivas propriedades CSS (*Cascading Style Sheet*), como altura, largura, cor de fundo entre outras propriedades relevantes para o *Web Developer*.

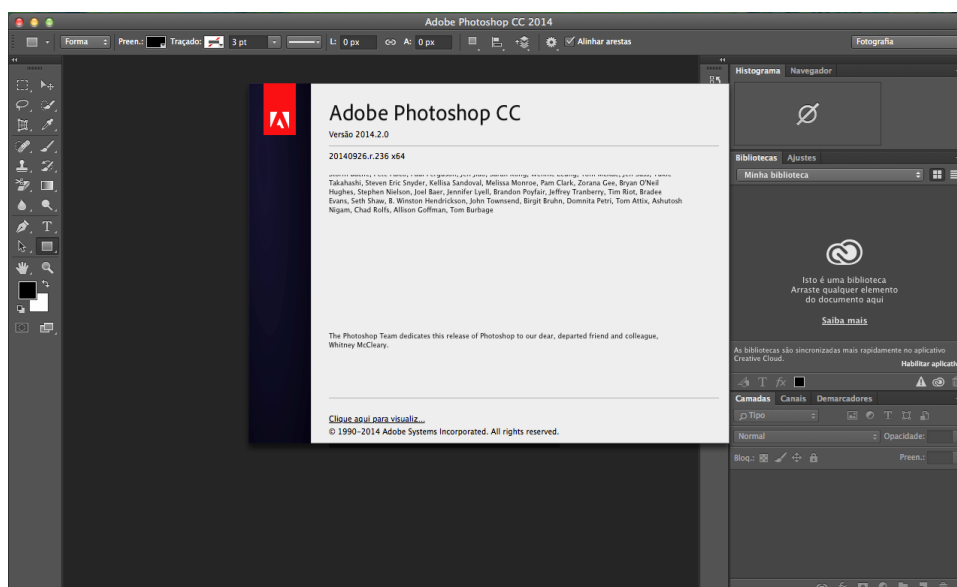


Figura 4 - Adobe Photoshop CC, versão Mac OS X

Adobe Photoshop, como Software pago, nesta nova versão, Adobe CC, existem os seguintes planos: fotografia e Aplicativo único.

No plano fotografia, é de realçar que além de vir adobe Photoshop, vem a aplicação Adobe Lightroom, que está disponível para qualquer plataforma, o preço deste plano é 9,99 Dólares por mês ou 119,98 dólares por ano.

No plano “Aplicação Única”, que permite somente escolher uma aplicação, terá os seguintes extras: 20 GB na Cloud, acesso ao *TypeKit*, etc, por 19,99 dólares por mês ou 239,88 dólares por ano. É de realçar que através da *Cloud* (Nuvem) é possível gerar um *Link* para partilhar [Creative Adobe, 2014].

Resumo:

- Importação: PNG, JPG, TIFF, BMP;
- Exportação / Guardar: PNG, JPG;
- Plataformas: Windows, Mac OS X;
- Custo:
 - “Fotografia”: 9,99 dólares por mês ou 119,98 dólares por ano
 - “Aplicação única” : 19,99 dólares por mês ou 239,88 dólares por ano
 - Trial de 30 Dias com 2 GB na *Cloud*;
- Website: <http://www.adobe.com>

GIMP (B)

GIMP, *GNU Image Manipulation Program*, é um programa gratuito, que além permitir o *Download*, sem qualquer custo associado, pode ser distribuído para qualquer pessoa. Outra característica, é de ser Open-Source, ou seja, qualquer pessoa pode ver código fonte, bem como contribuir para seu melhoramento. Contrariamente ao Software pago, que é desenvolvido por um conjunto de programadores bem remunerados, dedicados somente a esse projeto. O GIMP é desenvolvido por uma pequena equipa de pessoas, que não são remuneradas, mas entusiásticas, programadores voluntários. Atualmente, encontra-se na versão 2.8, disponível para os principais sistemas operativos, como Windows e Unix-Based, como Mac OS X e maioria das distribuições Linux. Esta ferramenta encontra-se bem documentada, em gimp.org, existindo livros e tutoriais publicados.

Quanto à interface, por omissão é constituída por 3 janelas flutuantes, chamadas de *multi-window mode* : *Toolbox*, *Layers* e área relativa à imagem. A primeira janela, *Toolbox*, contém ícones para as ferramentas mais importantes de seleção, de transformação e pintura. Na segunda janela contém as *Layers*, os canais de cor (RGB) e o histórico das modificações.

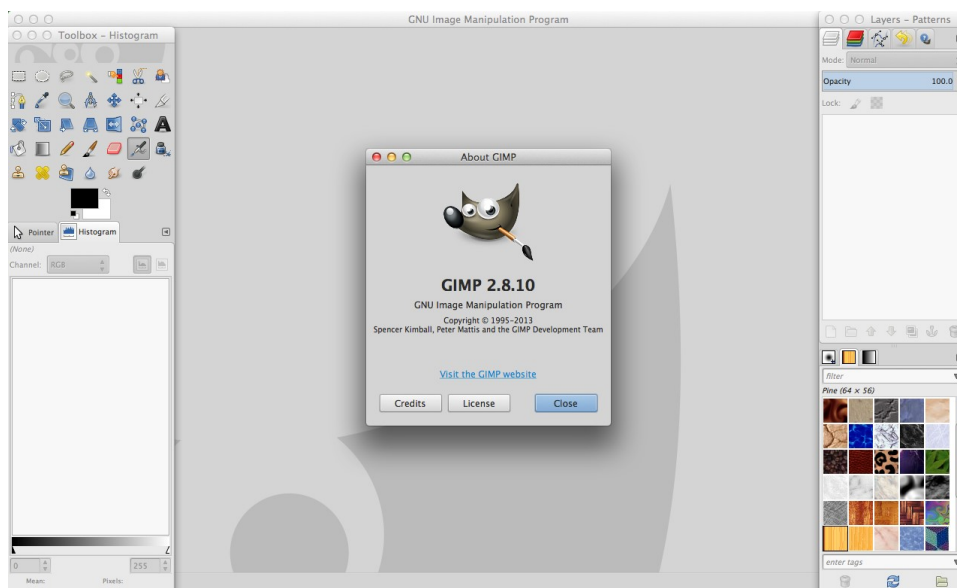


Figura 5 - Interface GIMP, versão Mac OS X

Na última janela ou secção, aparecerá a imagem selecionada e as modificações feitas pelo utilizador. Estas podem, facilmente, ser alteradas de acordo com o utilizador, acrescentado, removendo e ou agregando janelas. Outra possibilidade, é transformar numa interface mais tradicional, o *Single Window-Mode*, cujas janelas ficam fixas nas laterais. Ao nível de funcionalidades, existem ferramentas de seleção com forma retangular, de eclipse ou livre. Ferramentas de transformação como rotação e escalar.

A nível de filtros vai desde variação de *Blur*, *Noise* e até mesmo permite criar filtros e executar filtros através da linguagem Python ou Script-fu (linguagem de *query* á base de dados GIMP) [GIMP, 2015].

Resumo:

- Importação: Raw images, JPEG, PNG, GIF, TIFF, BMP, PDF, ICO;
- Exportação: JPEG, PNG, GIF, TIFF, BMP, ICO;
- Plataformas: Windows, Unix-Based (Mac OS X e Linux-Based desktop);
- Custo: gratuito;
- Website: <http://www.gimp.org>

Paint.Net (C)

Paint.Net é um editor de imagem gratuito para o sistema operativo Windows, atualmente na versão 4.06.

Após a versão 4, foi alterada a interface gráfica, mais concretamente o menu principal, simplificando certos menus e submenus. Mais concretamente janelas flutuantes Camadas, ferramentas e paletes de cores, ficaram agrupados em ícones, no lado direito da aplicação, para libertar área de trabalho.

Ao nível das funcionalidades, contém a usuais neste tipo de programas, tendo em conta que esta aplicação tinha como um dos principais objetivos fornecer, aos utilizadores mais avançados do sistema operativo Windows, uma melhor alternativa ao Microsoft Paint, no que toca a edição e tratamento de imagens, ferramentas de seleção e recorte, camadas, paleta de cores e transformações.

No que diz respeito a filtros ou manipulação da imagem, tem filtros para tratamento a nível de brilho, de Ruído (*Noise*), sépia e efeitos variados, como de distorção (*Blur*), Artístico (*Pintura a óleo*), Fotografia (por exemplo corrigir os olhos vermelhos).

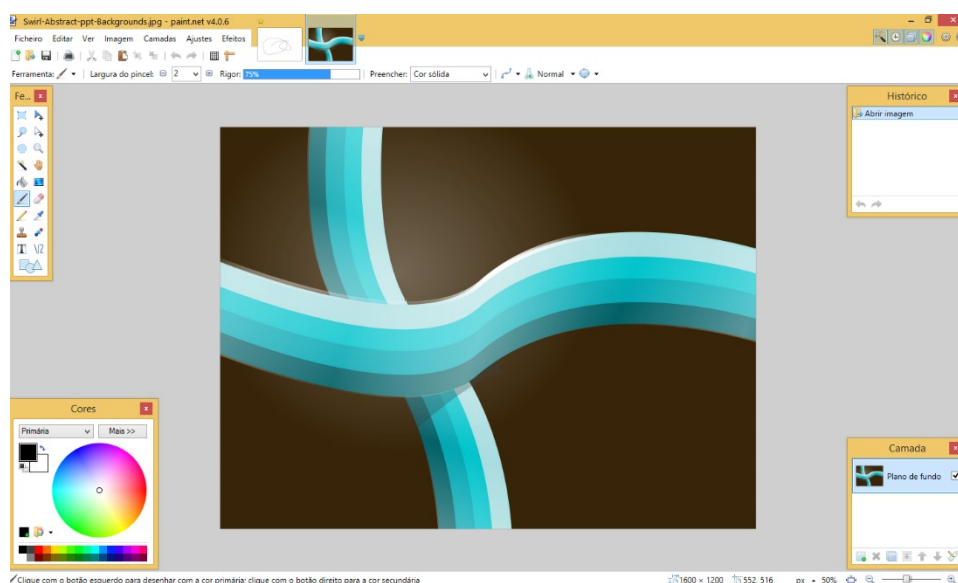


Figura 6 - Interface Paint.Net

Outra funcionalidade é o facto de permitir trabalhar com camadas, o que facilita no tratamento e manipulação da imagem.

Quanto aos formatos de imagens suportado, é possível abrir imagens dos formatos ditos normais como PNG, JPEG e BMP, bem como ICO e PDF, Quanto a gravar, permite guardar tanto no formatos mais comuns bem como no formato do próprio editor, conservando as propriedades ou funcionalidades de organização, como as camadas, intactas [Nick Peers, 2015].

Resumo:

- Importação: PNG, JPG, ICO, BMP;
- Exportação / Guardar: PNG, JPG;
- Plataformas: Windows;
- Custo: Gratuito;
- Website: <http://www.getpaint.net>

SumoPaint (D)

SumoPaint, é um programa de edição imagem, desenho e pintura,

A versão Online, utiliza a tecnologia Adobe Flash, é gratuita e não necessita de registo. Existe também, uma versão paga com o nome de SumoPaint Pro, com um valor de 4 dólares por mês, permitindo acesso todas as funcionalidades existentes, bem como acesso a uma versão para desktop.

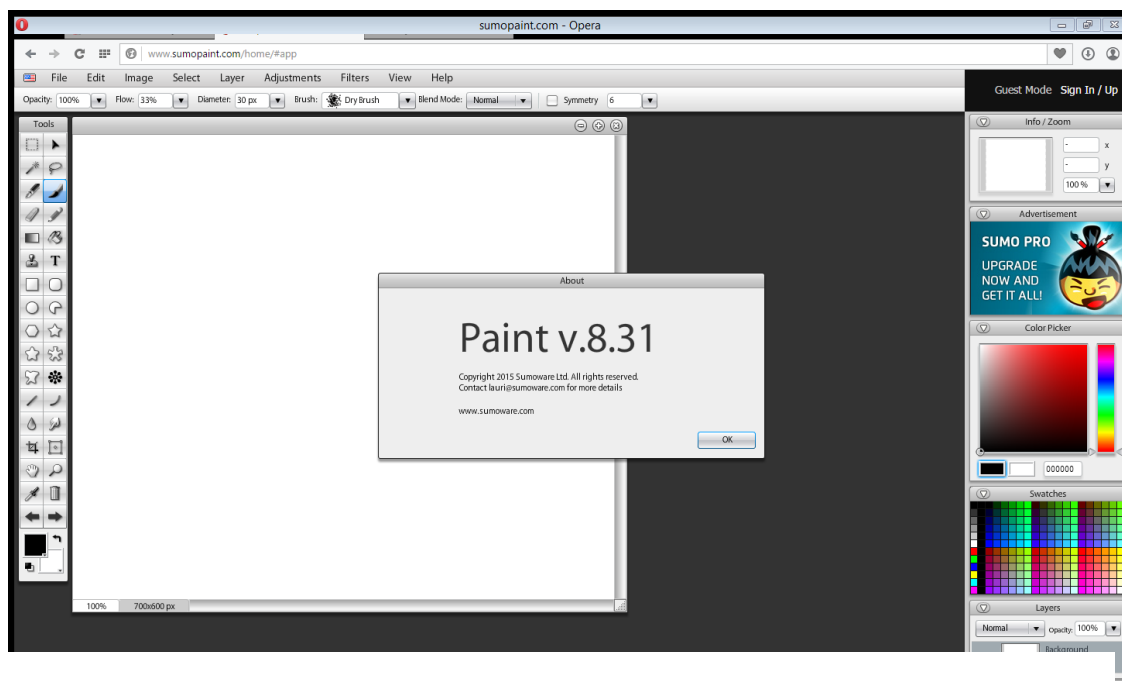


Figura 7 - Interface Sumo Paint

Ao nível de interface é muito semelhante a aplicações como GIMP e Adobe Photoshop. Quem está familiarizado com as duas ferramentas mencionadas anteriormente, não terá qualquer dificuldade de adaptar-se a esta aplicação, visto ser uma interface mais tradicional, tendo ferramentas e filtros tão bem conhecidos, ferramentas de seleção, de ajustamento de brilho, contraste e cor. Inclui também o conceito de *Layers* (camadas), variedade de *Brushes* (pinceis), gradientes e filtros 3D [Anthony Godinho, 2014].

Resumo:

- Importação: JPEG, PNG, sumo (formato próprio);
- Exportação: JPEG, PNG, sumo (formato próprio);
- Plataformas: Online (Adobe Flash), Windows;
- Custo:
 - online gratuito (limitado);
 - Versão Pro: 4 dolares /mês;
- Website: <http://www.sumopaint.com/home/>

AutoDesk Pixlr (E)

Pixlr, é um produto da Autodesk, composto por quatro aplicações. A primeira aplicação a ser lançada, em meados de 2011, dá-se pelo nome de Pixlr O-Matic. [Shaan Hurley, 2012], com o propósito de ser uma alternativa ao Instagram [Hélder Ferreira, 2013]. Esta aplicação veio com uma vasta variedade de filtros, permitindo aplicar vários em simultâneo. Além de aplicar filtros, esta aplicação inicialmente destinada para dispositivos móveis como Android e iOS, alargada posteriormente para plataforma web, utilizando a tecnologia Adobe Flash, permite partilhar as imagens via Email ou por SMS (*Short Message Service*).

Posteriormente, a Autodesk tirou partido do que melhor tinha o Pixlr-O-Matic juntamente com o conceito de editores como Adobe Photoshop. Nascendo a Pixlr Express, onde além dos filtros existentes na aplicação, foram adicionadas funcionalidades como *Crop* e rotação. Na funcionalidade de partilha, permite enviar para as redes sociais, tais como Facebook (no caso de dispositivos móveis, somente poderá enviar para as que estiverem instaladas), bem como via Bluetooth ou até mesmo através de serviços Nuvem (Cloud): Dropbox e Google Drive [Hélder Ferreira, 2013]. Esta aplicação, bem como a anterior, estão disponíveis para dispositivos móveis e versão web, sendo que esta última utiliza a tecnologia Adobe Flash.

Pixlr Editor, é uma versão mais tradicional da Autodesk, é uma reminiscência quer do Adobe Photoshop bem como do Paint.Net, que para utilizadores habituados a interfaces mais tradicionais existentes no mercado, facilita muito a sua adaptação.

Esta versão vai muito mais além das aplicações anteriormente mencionadas. Além de ter funcionalidades como *Crop* e rotação, contém um vasto conjunto de filtros, ferramentas de tratamento da imagem entre outras funcionalidades, sendo que este editor está somente disponível online, como os anteriores através da tecnologia Adobe Flash [Jack Dove, 2014].

Por último, a mais recente aplicação Autodesk Pixlr (for desktop), a Autodesk resolveu expandir o Pixlr da web e dos sistemas móveis para o desktop (Mac OS X e Windows). O conceito da interface é exatamente o mesmo das duas aplicações mencionadas anteriormente. Absorveu as mesmas funcionalidades, desde melhoramentos ao nível de contrastes até a aplicação de filtros.

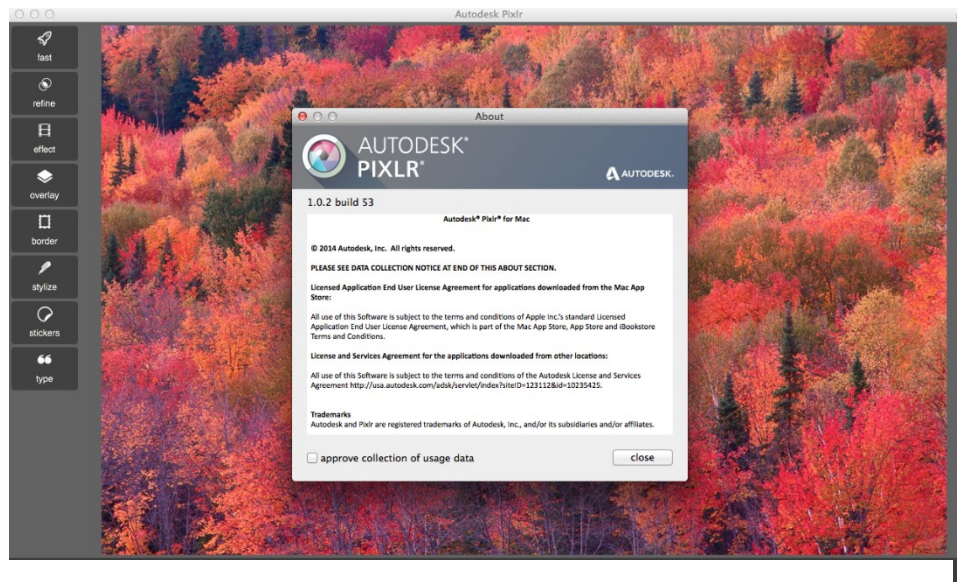


Figura 8 - Interface Pixlr, versão Mac OS X

À medida que se vai percorrendo a aplicação, a execução ou pré-visualização de algumas das funcionalidades não estarão, no imediato, disponíveis. Como por exemplo, ao mostrar a listagem de filtros existentes, caso alguns dos filtros ainda não tenha sido usado. Aparecerá um ícone de uma nuvem com uma seta para baixo, indicação que será necessário descarregar, onde esse requisito é relativamente rápido [Pedro Simões, 2014]. Quanto à partilha, está limitada a redes sociais Facebook e Twitter. A possibilidade de enviar por Email, onde caso o utilizador tenha algum instalado e configurado, automaticamente colocará a imagem alterada.

Todas estas aplicações, inicialmente são gratuitas, divide-se em três pacotes ou versões: *Starter*, *Essentials*, *Pro*.

Starter estará limitado ao número de filtros existentes bem como outras funcionalidades.

A versão *Essentials*, para além de aumentar o leque de funcionalidades e filtros, obriga ao utilizador a registar no Website da empresa, sem qualquer custo associado.

Já versão *Pro*, para além de ter acesso total as funcionalidades criadas pela empresa e obrigar ao utilizador registar-se, permite guardar as fotos na *Cloud* da empresa, o custo associado é de 1.99 dólares por mês ou de 14.99 dólares por ano. [Autodesk Pixlr, 2015].

Resumo:

- Importação: JPEG, PNG, GIF, TIFF, BMP;
- Exportação: JPEG, PNG, GIF, TIFF, BMP, pxd (imagem pixlr sobreposta);
- Plataformas: Windos, Mac OS X, Android, iOS, Online (Adobe Flash);
- Custo:
 - Starter: gratuito;
 - Essentials: gratuito mas necessita de registar-se;
 - Pro: 1.99 dólares / mês ou 14.99/mês;
- Website: <https://pixlr.com>

Fotor (F)

Fotor é um editor de imagem que existe em várias plataformas, utilizando em todas elas, interfaces diferentes das tradicionais, bonitas e simples de trabalhar.

Permite carregar várias imagens, trabalhando individualmente cada uma delas, fazendo *upload* através do computador do utilizador ou tendo uma conta em Dropbox, da rede social Facebook ou até mesmo através de um URL, onde os formatos tanto carregamento e gravação são PNG e JPEG.

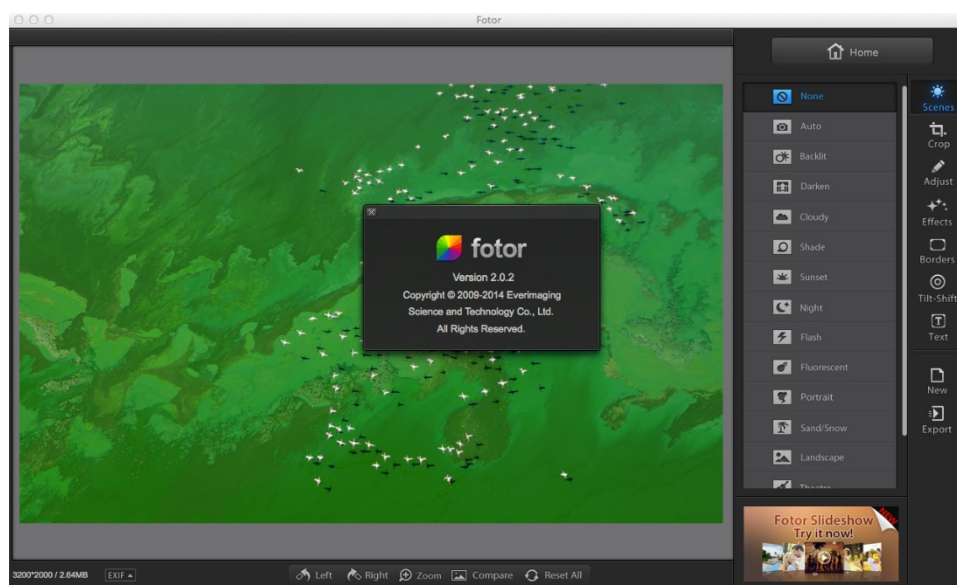


Figura 9 - Interface Fotor, versão Mac OS X

A nível de ferramentas, tem as comuns como *Crop*, *Resize* e *Rotate*. Quanto a filtros ou efeitos, existe menos convencionais ou conhecidos como *tiles*, que transforma a imagem pedaços retangulares.

Outra das funcionalidades interessantes, é o facto de permitir criar ou transformar a imagem numa moldura (*frame*), através de várias transformações ou modificações disponibilizadas.

Para além da plataforma online, existe também para desktop: Windows 8, Mac OS X. Bem como para dispositivos móveis: Android, iOS e Windows Phone /Mobile. Onde as funcionalidades poderão ser a mesmas ou menores consoante a plataforma ou dimensões dos dispositivos.

É de realçar que certos filtros ou funcionalidade, requerem registo, apesar de se poder fazer uma pré-visualização da funcionalidade, antes de a aplicar definitivamente à imagem, caso não tenha feito o registo e conseqüente o login [Norman, 2013].

Resumo:

- Importação: PNG, JPG;
- Exportação / Guardar: PNG, JPG;
- Plataformas: online (Adobe Flash), Desktop (Windows 8/8.1, Mac OS X), Smartphones (Android, iOS, Windows Phone/Mobile);
- Custo: Gratuito;
- Website: <http://www.fotor.com>

Picozu (G)

Picozu é um editor de imagens Online, gratuito, que tem por base a tecnologia HTML5.

Esta aplicação tem uma interface gráfica interessante, que permite ao utilizador alterar o seu *Workspace*. Oferece as seguintes opções: *minimal*, *Standard*, *Educacional* e *Kids*, onde a que se mais realça mais é o *Workspace Kids*. Este muda completamente o seu aspeto visual, mas não estrutural, ficando mais colorido e com um fundo diferente. Ainda relativamente à interface, o menu é mais reduzido comparativamente com os menus de editores tradicionais como Adobe Photoshop ou GIMP.

Ao nível de funcionalidades, carrega os formatos de imagens padrão, como PNG, JPG, BMP, até PSD (formato de Adobe Photoshop) e SVG.

Os formatos que permite guardar / exportar, além das mais comuns, ICO, PDF, TIFF e PS (Adobe Postscript).

Possibilita através de *tabs* trabalhar em várias imagens em simultâneo e transferir para as seguintes *Clouds* (Nuvem): Dropbox, Google Drive e Microsoft OneDrive. Quanto a redes sociais, permite fazer partilha somente para Facebook.

No que diz respeito a ferramentas (*toolbars*), encontra-se as usuais como seleção, recortar (*crop*), borracha, entre outras.

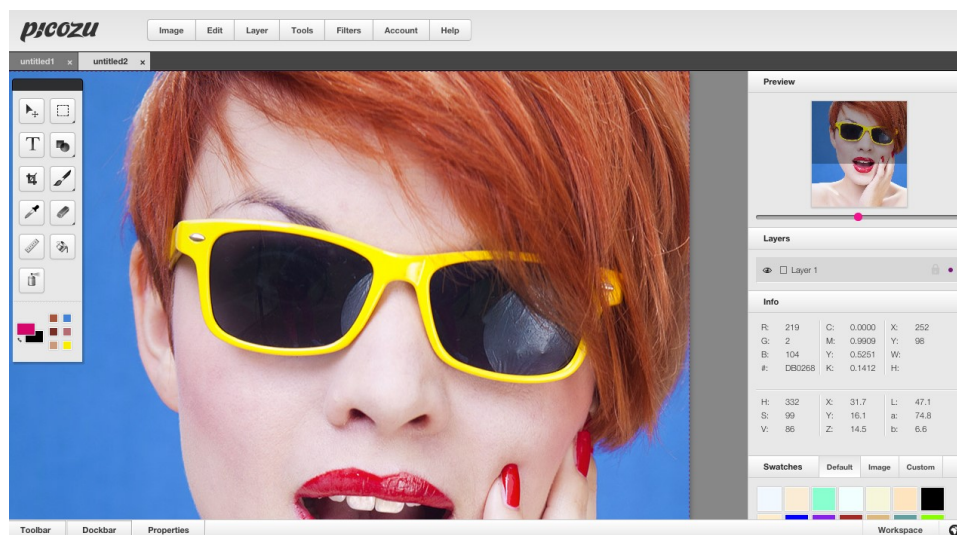


Figura 10 - Interface Picozu

No lado direito, figura 10, permite gerir *Layer* (Camadas), *Preview*, que permite fazer zoom in, zoom out á imagem seleccionada.

A nível de Filtros, onde existe uma boa organização, contém filtros de *Blur*, *Noise*, brilho, distorção, entre outras [Devicic, 2011].

Resumo:

- Importação: PNG, JPG, BMP, PSD e SVG;
- Exportação / Guardar: PNG, JPG, BMP, PSD, PDF, ICO, TIFF e PS;
- Plataformas: online (tecnologia base: HTML5);
- Custo: Gratuito;
- Website: <https://www.picozu.com>

2.1.2 Análise Comparativa

Nesta secção é realizada uma análise das aplicações anteriormente descritas onde as letras do alfabeto estão associadas á ordem em que as aplicações foram apresentadas:

A) Photoshop; B) GIMP; C) Paint.Net; D) Sumo Paint; E) Pixlr; F) Fotor; G) Picozu.

2.1.2.1 Importação e Exportação

Esta secção, diz respeito à maneira como o utilizador poderá ir buscar a sua imagem bem como poderá guardar o seu aspeto final. **I** a significa importação e **E** a Exportação. A maioria do Software apresentado carrega e grava diretamente, a imagem no disco do utilizador. Aproximadamente mais de metade possibilita importação da imagem via URL.

Dois programas conseguem carregar e posteriormente gravar a imagem através da conta Facebook, e gravar nessa conta. Apenas uma consegue carregar e gravar imagem através de uma conta Nuvem, neste caso, Dropbox.

Também somente uma consegue, caso o utilizador pretenda, capturar imagem através da câmara tanto incorporada tanto portátil como num smartphone.

Em suma, as ferramentas que oferecem mias opções ao utilizador são Picozu e Autodesk Pixl, estando a primeira disponível via web e a segunda para várias plataformas.

Tabela 1 - Forma de Importação e Exportação

	A	B	C	D	E	F	G
Disco	I & E	I & E	I & E	I & E	I & E	I & E	I & E
URL		I		I	I		I
WebCam					I		I
DropBox							I & E
Facebook					I & E	E	I & E

2.1.2.2 Formato de Importação e Exportação

Relativamente à área de tipos de imagens que os utilizadores podem utilizar e que formato pode guardar as alterações, no quadro abaixo são apresentado dois formatos mais usuais: JPEG e PNG, em que todos possibilitam quer a ler como aguardar nesses dois formatos.

O formato que somente dois editores suportam é ICO, sendo que, a solução mais completa no que diz respeito ao suporte de formatos de imagem aqui apresentados é o Picozu.

Tabela 2 - Formato de Importação e Exportação

	A	B	C	D	E	F	G
JPEG	I & E	I & E	I & E	I & E	I & E	I & E	I & E
PNG	I & E	I & E	I & E	I & E	I & E	I & E	I & E
BMP	I & E	I & E	I & E		I & E	I & E	I & E
TIFF	I & E	I & E	I & E		I & E	I & E	I & E
ICO		I & E					I & E

2.1.2.3 Funcionalidades

Quanto às funcionalidades descritas, a maioria das funcionalidades selecionadas e importantes para utilizador. Realça-se a aplicação Picozu, a única aplicação online, tem como base a tecnologia HTML5, a fornecer a todas a funcionalidades.

Tabela 3 - Funcionalidades

	A	B	C	D	E	F	G
Recortar	✓	✓	✓	✓	✓	✓	✓
Redimensionar	✓	✓	✓	✓	✓	✓	✓
Camadas	✓	✓	✓	✓			✓
Texto	✓	✓	✓	✓	✓	✓	✓
Fontes Locais	✓	✓	✓		✓	✓	
Fontes Web	✓			✓	✓	✓	✓
Filtros	✓	✓	✓	✓	✓	✓	✓
Ajustamento							
Filtros de Ruído	✓	✓	✓	✓			✓
Filtros de Distorção	✓	✓	✓	✓			✓
Efeitos	✓	✓	✓		✓	✓	✓
Molduras					✓	✓	✓

2.1.2.4 Partilha

Hoje em dia as redes sociais, fazem parte do quotidiano do utilizador, como sendo necessário que as aplicações estejam interligadas entre elas é fundamental. As três últimas ferramentas a serem descritas têm acesso online.

Tabela 4 - Partilha

	A	B	C	D	E	F	G
Facebook					✓	✓	✓
Twitter					✓	✓	
URL						✓	

2.1.2.5 Plataformas

A maioria das aplicações encontra-se em ambiente Offline. Dessa maioria, está também disponível online. Destacando-se Autodesk Pixlr, que curiosamente, foi disponibilizada para *Smartphones*, posteriormente Online e mais recentemente para os sistemas operativos mais utilizados, Windows e Mac OS X.

Tabela 5 - Plataformas

	A	B	C	D	E	F	G
Windows	✓	✓	✓		✓	✓	
Windows Phone						✓	
Mac OS X	✓	✓			✓	✓	
iOS					✓	✓	
Linux		✓					
Android					✓	✓	
Online				✓	✓	✓	✓

2.1.3 Conclusão

Foram apresentadas e descritas várias ferramentas no âmbito de edição e manipulação de imagem existentes no mercado, sendo a maioria delas gratuitas, o que significa que existem no mercado boas alternativas. Posteriormente foi feita uma análise comparativa que abrangeu o nível de formatos suportados, o meio de importação e exportação, as funcionalidades mais relevantes, a forma de partilha mais simples e rápida, bem como as plataformas para as quais acessíveis.

Uma das evidências na análise das ferramentas é que em mais de metade, das versões web, a tecnologia utilizada é o Flash (através da linguagem ActionScript 3) da Adobe. Também se verificou, somente uma delas utiliza a tecnologia HTML5: o Picozu, sendo esta que se destacou em quase todos quadros de análise.

Em termos de interface gráfica, durante análise aquelas que mais realçam aos olhos do utilizador pela sua forma diferenciadora das interfaces tradicionais deste tipo de aplicações são Autodesk Pixlr e Fotor. Graças à simplicidade e boa organização, qualquer utilizador adapta-se bem e rápido.

Os vários editores de manipulação e edição aqui apresentados permitem ter uma visão geral, não só da oferta existente no mercado, como também das funcionalidades que serão implementadas no projeto e no cuidado a ter na interface gráfica, em termos de usabilidade.

2.2 Tecnologias Web

Nesta secção são apresentadas as possíveis tecnologias Web a serem utilizadas no projeto, bem como as Frameworks ou bibliotecas que podem acelerar o desenvolvimento.

2.2.1 Adobe Flash

2.2.1.1 Breve História

Começou por ser um Software direcionado para desenho e ilustrações com o nome de SmartSketch. Foi Criado por Jonathan Gay e Charlie Jackson, em 1994, proprietária da empresa FutureWave que fundou em 1993. Mas, devido à forte concorrência de Software como Illustrator e FreeHand, mantendo as funcionalidades já criadas e com feedback de pessoas próximas, bem como pesquisa na internet, decidiram adicionar funcionalidades dedicadas a animação. Em 1995, já com a incorporação de funcionalidade de animação, mudaram para o nome para CelAnimator. Em 1996, mudam novamente o nome para FutureSplash Animator, que permitia, além desenho vetorial, animar esse mesmo desenho e criar botões, para ativar essas mesma animações, bem como permitia incorporar em página HTML [Yassine Bouhleb, 2010]. Houve grande sucesso, devido ao plug-in ser muito compacto bem como os ficheiros gerados, numa altura em a Internet era muito limitada. Por outro lado chamou a atenção da Macromedia ShockWave, que cooperava em muitos projetos com a Disney, esta ultima cliente da FutureWave.

A Macromedia convidou a Jonathan Gay e Charlie Jackson, para trabalharem juntos e, em Dezembro de 1996, compra FutureWave Software e muda o FutureSplash Animator para Macromedia Flash 1.0. O nome Flash é contração entre as palavras Future e Splash. Em 2005, a Adobe compra a Macromedia e torna-se Adobe Flash, que é incorporado na suite Adobe CS3, lançada em 2007 [Christina Warren, 2012].

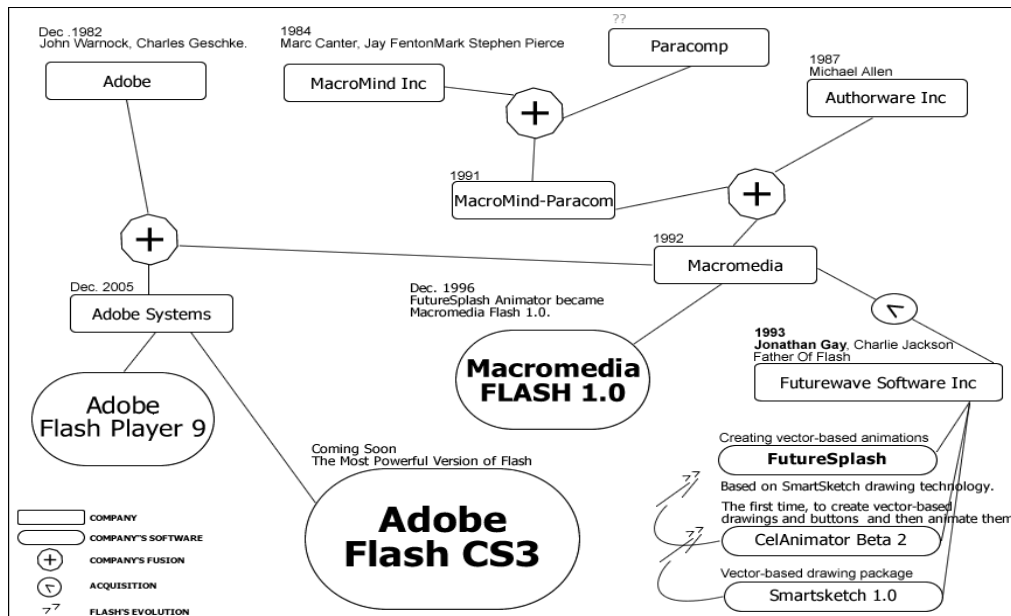


Figura 11 - A quick tour of Flash's Genealogy Tree [Yassine Bouhiel, 2010]

Hoje em dia é muito utilizado para visualização de vídeos, *Streaming*, publicidade, entre outros. Para esta tecnologia funcionar no lado cliente (browser), é necessário a instalação de *Plugin* no Browser ou *Player* no computador do utilizador. Como linguagem de programação é utilizado ActionScript, versão 3.

Em junho de 2015, a Google, anuncia que na próxima versão do seu Browser, desativaria ou “bloquearia”, por omissão, conteúdos Flash, visto consumir recursos, para salvaguardar as baterias dos portáteis, onde para já foi implementada na versão 44 beta, neste e lançado mesmo mês [Pedro Pinto, 2015].

Não só problemas de consumo de recursos excessivo padece esta tecnologia, vindo a público problemas de segurança, principalmente nos primeiros 6 meses deste ano, 2015. A titulo exemplo, dados roubados de uma empresa de segurança, a Hacking Team, revelou que existiam, na versão 18.0.0.204, duas falhas de segurança. Cujas, possibilitavam a execução de código malicioso após o Flash apresentar problemas e deixar de funcionar [Pedro Simões (2), 2015].

Na sequência dos problemas de segurança anteriormente referidas, a Mozilla, fundação responsável pelo Browser Firefox, chegou a suspender o suporte ao Flash até que os erros de segurança fossem resolvidos o mais rapidamente. Em paralelo a esta situação, pessoas como, o chefe de segurança do Facebook e responsável de suporte do Firefox, relembrou palavras proferidas por Steve Jobs (foi co-fundador, presidente e diretor executivo da Apple Inc) para justificar o fim da tecnologia Flash e a adoção da tecnologia HTML5 [Pedro Simões (3), 2015].

2.2.1.2 Enquadramento em Dispositivos Móveis

Contrariamente aos dispositivos desktop, que tiveram e continuam ter, apesar de problemas de segurança, uma integração e utilização massiva desta tecnologia, nos dispositivos móveis o cenário difere, muito por causa dos responsáveis pelos respetivos sistemas operativos móveis.

Um dos primeiros, a mostrar essa tendência de mudança veio foi a Apple, mais concretamente por Steve Jobs em 2010, à altura responsável máximo pela Apple.

No texto publicado no Website da Apple, em abril de 2010, pelo próprio Steve Jobs, aponta algumas razões pelas quais esta tecnologia não era compatível com o seu sistema operativo móvel [Steve Jobs, 2010]. Além de se referir a Adobe como empresa, salienta que Adobe Flash é uma tecnologia fechada, onde não é possível correr jogos no iPhone através do Flash. Realçando, no entanto, o facto de existirem mais de 50 000 jogos disponíveis na Apple Store. Também salientou, segundo Symantec classificou, a tecnologia Flash como uma das piores em 2009 a nível da segurança.

O Flash não tem uma boa performance pois, aquando da descodificação de vídeo por Software, por parte do Flash, diminui o tempo de utilização das baterias.

Por último indica que esta tecnologia, foi criada na era dos computadores que utilizam teclado e rato logo, não estava preparada para eventos Touch.

O futuro passava por dispositivos móveis com *Touchscreen*, neste sentido não é viável o Adobe Flash.

Em jeito de conclusão, Steve Jobs, indicava que nesta era Mobile, passava pela utilização de Web Standards abertos e que a Adobe devia-se focar na criação de ferramentas para utilização do HTML5.

Em 2011, a Adobe anuncia que não implementará o *Plugin* para os Browsers, nos sistemas operativos Android a partir da versão 4.1. A própria Adobe fez a recomendação de desinstalar o Flash Player, aqueles dispositivos que terão acesso à nova versão do Android, bem como não seria disponibilidade na loja oficial Google Play [Victor M., 2012], confirmando assim o fim do desenvolvimento do *Plugin* para sistemas móveis.

2.2.2 HTML

2.2.2.1 Breve História

A origem do HTML, deriva da Markup Language chamada SGML (Standard Generalized Markup Language). O protótipo do HTML (HyperText Markup Language), criado por Tim Berners-Lee em 1992, tem como objetivo ligar vários documentos através de hipertexto. Do desenvolvimento do HTML, nasceu o protocolo para aceder somente não a um, mas outros documentos de texto através de *Hyperlinks*. Onde este protocolo foi batizado HTTP (Hypertext Transfer Protocol) [shenoy et al, 2013].

A *Standardization* (padronização) é um processo contínuo. Várias modificações e versões foram criadas com a seguinte cronologia onde W3C (*World Wide Web Consortium*), fundada em 1994 pelo mesmo Tim Berners-Lee [W3Schools, 2015]:

- HTML 2.0 (Novembro de 1995)
- HTML 3.2 (Janeiro de 1997)
- HTML 4.0 (Dezembro de 1997)
- HTML 4.01 (Dezembro de 1999)

Como reformulação do HTML4 em XML (*eXtensible Markup Language*), surgiu em 2000, o XHTML 1.0 com recomendação da W3C (*World Wide Web Consortium*), impondo regras mais rigorosas. Obrigava, por exemplo, que as *tags* (elementos) fossem somente em letras minúsculas e fechadas, incluído *tags* vazias como tag de imagem (). Com estas regras de certa forma passou a ser necessário escrever HTML com melhor qualidade.

Em 1996, foram lançadas duas tecnologias, importantes atualmente: DOM (*Document Object Model*) e JavaScript, com a nova versão Netscape Navigator, versão 2.0. Hoje em dia projecto á parte da Netscape, dentro Mozilla [Keith et al, 2010]. Normalmente fala-se das duas tecnologias como um só, o que não é verdade porque cada uma tem a sua missão. DOM é a API (*Application Programming Interface*) responsável pela interação com os elementos do documento HTML, enquanto JavaScript é a linguagem responsável pela interação com essa mesma API [Tiffany B. et al, 2014].

Em 1998, estas tecnologias foram normalizadas com a recomendação por parte da W3C [W3C, 2015].

Em 2004, a Mozilla e Opera apresenta alguns conceitos ou propostas para evolução de HTML, a nível de *Web Forms*, à W3C. Essa proposta de alguns foram rejeitadas por parte da W3C, por divergências sobre a direção a tomar estaria ou seria a mais correta [W3C (b), 2015].

A W3C, escolheu criar e desenvolver, a evolução, com base na tecnologia XML. Posteriormente a Mozilla, Opera e por último a Apple, anunciaram em conjunto a criação de comunidade de WHATWG (*Web Hypertext Application Technology Group*) [WHATWG, 2015], focando-se na evolução do HTML bem como da API que preenchem as necessidades das aplicações Web [WHATWG (b), 2015].

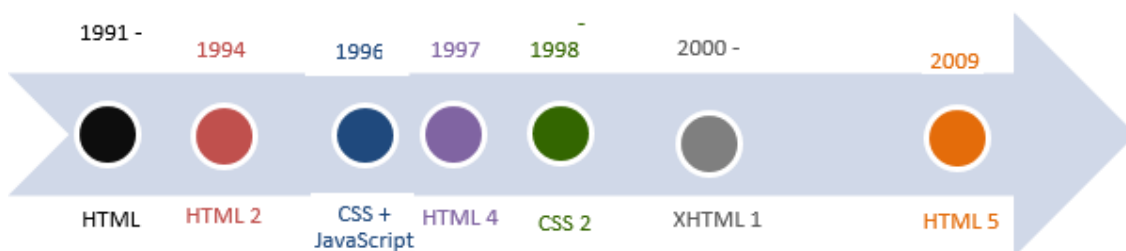


Figura 12 - Cronologia das tecnologia HTML e CSS

Em 2006, W3C mostra interesse em participar no desenvolvimento da especificação do HTML5 e cria em 2007 uma equipa à parte para integrar na WHATWG e participar no desenvolvimento dessa mesma especificação.

Até que em 2011, os dois grupos chegaram à conclusão que estavam divergir no ponto de chegada. Ou seja, enquanto a WHATWG tem a filosofia de “*Living Standard*”, desenvolvimento contínuo ao longo do tempo, adicionando novas funcionalidades à medida que for necessário, sendo filosofia mais aberta, já a W3C tinha filosofia de publicar uma “versão final do HTML5”, que foi pública já com W3C recomendação em 28 de Outubro de 2014 [W3, 2015].

2.2.2.2 HTML5 – Evolução do HTML

A base do HTML5 provem de duas especificações que foram canceladas: Web Forms 2.0 (por parte da W3C) e Web Apps 1.0 (por parte WHATWG) [Goldstein et al, 2015].

O termo HTML5, refere-se, para além de certas melhorias, a um conjunto variado de novas tecnologias web e API [Goldstein et al (b), 2015].

Para se entender melhor as facetas do HTML5, no ponto de vista de Mathias Bynens [Mathias Bynens, 2010], divide em três níveis ou camadas, o processo de especificação do HTML5.

O primeiro nível: simplificar o que já existe. Refazer certas tags (elementos) da versão anterior do HTML, o HTML4. Coloca, por exemplo, como opcional certos atributos, deixando de ser necessário na *tag Link* ou script, referir o tipo de ficheiro (*type*); *tag meta* deixa de ser necessário o atributo content e mesmo não sendo obrigatório fechar com “/” as *tags* vazias como de imagem (*img*).

A segunda camada: especificar e estruturar melhor. Enquanto na camada anterior não afeta quem tem um Browser mais antigo, esta camada é compatível apenas nos Browsers mais

recentes. O autor refere a novos valores para o atributo *type* da *tag input*, como por exemplo: *email*, *tel* (para telefone), *number* (para números genericamente), *url* (para endereços de internet), *datetime*, *date*, etc. E também novos atributos para o elemento *input* como *placeholder* e *autofocus*.

Nesta segunda camada, incluo contrariamente à visão do artigo, de Mathias Bynens, os novos elementos, chamados de *sectioning elements* (elementos de estruturação ou semântica): *header*, *nav*, *section*, *article*, *footer*, *aside* e *main*, a fim de impedir o abuso da tag *div*. Com o propósito código mais limpo e obter maior acessibilidade, sobre tudo ajuda na classificação do Website, nos motores de pesquisa no chamado SEO (Search Engine Optimization) [shenoy et al (b), 2013].

Outros elementos como *mark*, *time*, *output* e *address*, estes além de ajudar no SEO, servem para substituir ou incentivar a diminuição do uso do elemento *span*, segundo Mathis Bynens.

A última camada: usar HTML5 na sua total capacidade. Apresenta elementos direcionados para interatividade / multimédia, usados exaustivamente através de JavaScript. Através de novas APIs JavaScript do HTML5, como vídeo, áudio, Canvas, *Drag & Drop*, SVG, WebStorage (LocalStorage e SessionStorage), IndexedDB, etc. Estes elementos, tal como acontece na segunda camada, só funcionam em Browsers mais recentes.

2.2.2.3 Vantagens e Desvantagens

Como todas as novas tecnologias, tem os seus pros e contras

Começando pelas suas vantagens, elas são as seguintes, com base no anteriormente referido:

- A desnecessária instalação de *Plugins*, podendo executar vídeo e som nativamente;
- Maior relevância em termos SEO;
- Código mais limpo;
- Maior interatividade com o utilizador com menos código;
- O facto de web app poder ser utilizado mesmo no caso de não ter acesso á web;
- A possibilidade de armazenamento, certos dados no lado do cliente (Browser);
- O facto não estar dependente a nível de processamento exclusivamente no lado do servidor. Podendo estar repartido entre lado do cliente (Browser) e o servidor.

Relativamente às suas desvantagens:

- A necessidade de ter sempre um Browser mais moderno e sempre atualizado;
- O facto de mesmo alguns dos Browser mais utilizados, não terem incluído toda a especificação do HTML5;
- Para o bem e para o mal, de cada vez mais se suportar na linguagem JavaScript.

2.2.2.4 SVG

Em 1998, a W3C, criou um grupo de trabalho para o desenvolvimento de representação gráfica vectorial como aplicação XML (*eXtensible Markup Language*). Com SVG (*Scalable Vector Graphics*), aplicação baseada em XML, a informação de uma determinada imagem é guardada num ficheiro de texto normal. Há vantagem de XML ser uma tecnologia aberta, portabilidade e interoperabilidade. Os programas de CAD (*Computer-aided design*) e desenho gráfico têm a possibilidade de importar ou exportar desenho em formato SVG. Que reforçou entre várias aplicações como sendo um formato Standard para troca de informação [Eisenberg et al, 2014]. Esta tecnologia em 2001, mais concreta a especificação do grupo criado, teve recomendação da W3C [W3Schools (b), 2015].

Anterior ao suporte para Web, SVG, com base na linguagem XML para criação de imagens vectoriais, apareceu com notoriedade em 2000. Com a Adobe a fornecer o primeiro suporte para Browser, via *Plugin*, mas também aplicações desktop, como CorelDraw e Microsoft Visio. Muitas aplicações têm um suporte estável hoje em dia para importar e / ou exportar imagens em formato SVG ou são baseado em desenho vectorial, como é o caso de Adobe Illustrator, CorelDraw, Inkscape, Scribus [Frost et al, 2012].

A nível de suporte em Browsers, feita nativamente, surgiu pelo Konqueror em 2004. No ano a seguir, em 2005, Opera suporta bem avançada, enquanto o Browser Mozilla Firefox integrava suporte somente básico desta tecnologia. Em Browsers com motor de JavaScript Webkit, como Safari, Google Chrome, etc, Incorporavam suporte em 2006, mesmo que ainda incompleto. Quanto ao Microsoft IE (Internet Explorer), somente a partir da versão 9 (2011) como o Firefox, incorporava suporte básico desta tecnologia, onde na versão seguinte, IE10, vinha a ter suporte mais extenso, implementando funcionalidades como filtros [Microsoft, 2014]

SVG Exemplos by http://www.w3schools.com/svg/svg_inhtml.asp



Figura 13 - SVG: Demonstração

Na figura acima, pode visualizar-se o desenho de formas geométricas, através de XML. SVG, como um formato de imagem, tem a vantagem de ser [W3Schools (c),2015]:

- Ser criado, editado em qualquer editor de texto;
- Os ficheiros são baseados em XML;
- Possibilidade de serem indexadas;
- Possibilita ser importado, exportado de e para qualquer programa de desenho vectorial;
- É um Standard aberto.

SVG, com integração no Browser mais concretamente, tem funcionalidades como [mooock.org, 2014]:

- Formas Básicas (Retângulos, círculos, Elipse);
- Texto (tamanho, espessura, cor);
- Transformações (rotação, translação, escalar);
- Gradientes (linear, radial);
- Filtros;
- Animações;
- Aplicar estilos com CSS e Fontes através de CSS.

2.2.2.5 HTML5 Canvas

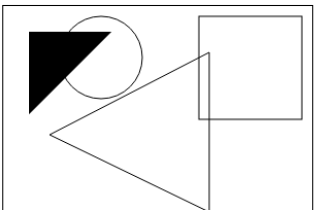
O elemento Canvas, teve a sua origem pelas mãos da Apple, mais propriamente da sua framework Quartz 2D, para desenho a duas dimensões [Adam Roberts, 2014]. Que na altura, o principal propósito era para criação de *widgets* para *Dashboard*, que aparece no próprio sistema operativo (Mac OS X) em *Mission Control*. Atenção de outros competidores como Mozilla e Opera, mais tarde foi acolhida pela WHATWG e conseqüente pela W3C [Rob Hawkes, 2011].

Aparece em Browsers através da Apple, no safari 1.3, mais tarde em Mozilla Firefox desde versão 1.5, Opera na versão 9 e desde a primeira versão no Google Chrome, ou seja, desde 2008.

Segundo os autores Steven e Jeff Fulton [Flaton, 2013], o elemento Canvas é uma área do ecrã que está em *“immediate mode Bitmapped”*, manipulada através de JavaScript. Através da Canvas API, é feita atualização de bitmap, constituído por pixéis, existente no elemento Canvas. Por último Canvas tem a possibilidade além de contexto 2d, utilizar a 3D através da API WebGL (*WebGL context*), mais concretamente baseado em OpenGL ES [MDN, 2015]

Na figura abaixo podem visualizar algumas capacidades do elemento Canvas juntamente com a sua API JavaScript:

Canvas ao estilo de desenho vectorial



Carregar imagem - Original





Imagem no elemento canvas



```
<!DOCTYPE html>
<html>
<head> ... </head>

<body onload="draw();">
  <h3>Canvas ao estilo de desenho vectorial</h3>
  <canvas id="tutorial" width="300" height="200"></canvas>
  <br/>
  <h3>Carregar imagem - Original</h3>
  
  <br/>
  <h3>Imagem no elemento canvas</h3>
  <canvas id="cv2"></canvas>
  <script>
    function draw() {
      var canvas = document.getElementById('tutorial');
      if (canvas.getContext) {
        var ctx = canvas.getContext('2d');
        // rectangulo
        ctx.strokeRect(190, 10, 100, 100);

        // triangle preenchido
        var ftriangle = new Path2D();
        ftriangle.moveTo(25, 25);
        ftriangle.lineTo(105, 25);
        ftriangle.lineTo(25, 105);
        ctx.fill(ftriangle);
        // triangulo sem fundo
        var striangle = new Path2D();
        striangle.moveTo(200, 200);
        striangle.lineTo(200, 45);
        striangle.lineTo(45, 125);
        striangle.closePath();
        ctx.stroke(striangle);
        //circulo
        ctx.beginPath();
        ctx.arc(95, 50, 40, 0, 2 * Math.PI);
        ctx.stroke();
      }
      //criação novo canvas onde vai ser desenhado
      var cv = document.querySelector('#cv2');
      var ctx2 = cv.getContext('2d');
      //Associa o mesmo tamanho da imagem ao canvas
      cv.width = document.querySelector("#im").width;
      cv.height = document.querySelector("#im").height;
      //Aparecerá a imagem no Canvas com picotado
      ctx2.drawImage(document.querySelector("#im"), 0, 0);
    }
  </script>
</body>
</html>
```

Figura 14 - HTML5 Canvas: Código Exemplo

Num primeiro exemplo, é demonstrado o desenho de formas geométricas, enquanto no segundo exemplo podem visualizar a forma como é desenhada a imagem no elemento Canvas, com o respetivo código no lado direito de cada exemplo executado no lado esquerdo.

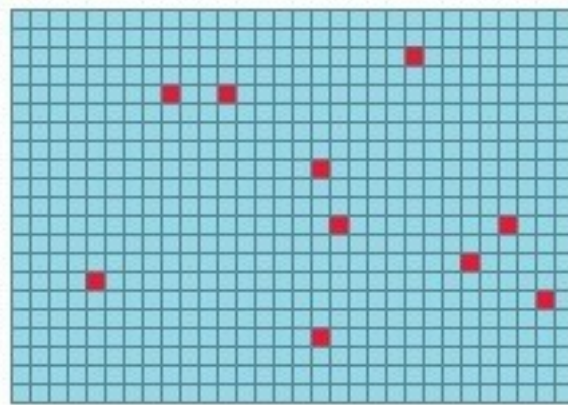
O Canvas através da sua API, permite com o contexto 2D, à semelhança do SVG [Tom Green, 2015]:

- Formas básicas: linhas, arcos, curvas quadráticas e curvas de bezier;
- Texto: tamanho de texto, preenchimento, cor, gradientes (Linear, radial);
- Imagens: carregar, criar, desenhar, escalar, cortar;
- Aplicar estilos: cor, preenchimento, transparência, sombra, gradiente;
- Transformações: translação, rotação, escalar, transformação por matriz;
- Animação: executar funções de desenho, associando intervalos de tempo;
- Eventos: teclado, rato, *Touch* (dedos no ecrã);
- Coordenadas;
- Reprodução de Vídeo.

2.2.2.6 SVG versus Canvas

Os dois principais sistemas de representação informação gráfica são: rastrear (bitmap) e vectorial.

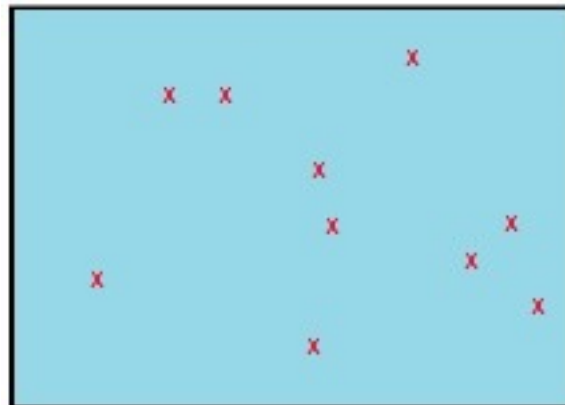
No tipo raster, a imagem é representada por pequenos pedaços retangulares de imagem ou pixéis. Onde cada Píxel é representado pelo valor RGB da cor ou como um índice dentro de uma lista de cores. A este grupo de pixéis, conhecido por bitmap, é normalmente guardado num formato comprimido. Devido a utilização nos ecrã mais modernos deste tipo de apresentação, apresentar a imagem requer programa de visualização para executar a descompressão do bitmap e transferir para o ecrã [Eisenberg et al (b), 2014].



Raster point features

Figura 15 - Imagem Raster / bitmap

No tipo vectorial, a imagem é descrita como uma sequência de formas geométricas. Contrariamente ao tipo raster, que recebe um conjunto de pixels, um programa de visualização vectorial recebe instruções para desenhar formas num conjunto de coordenadas. Outra forma de descrever é a representação de uma grelha de pontos, onde as linhas e curvas são para ser desenhadas [Eisenberg et al (c), 2014].



Point features

Figura 16 - Imagem vectorial

As duas tecnologias anteriormente descritas, Canvas e SVG, utilizam cada um destes tipos de sistemas gráfico. Canvas utiliza sistema bitmap, enquanto SVG utiliza o sistema vectorial, conforme se pode verifica na tabela abaixo [Rob Crowther et al, 2014].

Tabela 6 - Diferenças entre sistemas gráficos

Tópico	Raster (Bitmap)	Vectorial
Ficheiros (formato)	.gif, .jpeg, .png	.svg, .ai, .eps
Criado através de	Pixéis	Equações Matemáticas
Criado em programas como	Photoshop, Gimp	illustrator, inkscape, Flash
Destinada a	Websites, fotografia, gráficos	Ícones, logos, animações
Tamanho do ficheiro	Grande	Pequeno
Utilização em 3D	Texturas	Objetos (Formas geométricas)

No entanto, cada uma destas tecnologias com os consequentes sistemas gráficos, no mundo real tem casos de uso distintos, que serão dados exemplo a seguir.

Normalmente, SVG é adequado [Tiffany B. et al (b), 2014]:

- Imagens estáticas, especialmente para *Layouts* adaptáveis e fluidos;
- Imagens redimensionadas, para qualquer dimensão sem perder qualidade;
- Projetos dos quais se beneficia pelos métodos do DOM para atribuição de eventos e manipulação de objetos;
- Projetos onde é criados imagem no lado do servidor;
- Projetos onde acessibilidade e SEO é importante.

No caso do Canvas [Tiffany B. et al (c), 2014]:

- Imagens bitmaps, edição de fotografia, vídeo ou qualquer operação que requeira manipulação de pixéis;
- Imagens no qual é necessário a criação e animação no comum;
- Em jogos.

2.2.3 CSS

2.2.3.1 Breve História

CSS (*Cascading Style Sheet*) é uma linguagem de estilos com o objetivo de descrever o aspeto e a formatação de um determinado documento. No caso específico de página HTML, a forma de simplificar o processo de desenvolvimento, como aplicação de fontes e cores, onde a partir do HTML4, é permitido definir os estilos em ficheiros separados [W3Schools (d), 2015].

Inicialmente, não havia CSS ou qualquer tipo de formatação no HTML, em que cada Browser tinha o seu próprio CSS e conseqüentemente formas diferentes de mostrar a mesma página. Esta situação elevou a preocupação dos programadores porque não tinham qualquer controlo sobre a maneira de formatar a página web. Para além de formatar e / ou descrever o aspeto a página, permite fazer a separação do que é o conteúdo do documento da apresentação [Emily, 2015].

Para tentar resolver o problema da desfragmentação de estilos, foi apresentado à W3C, recentemente fundada, uma proposta de *Standard* em 1994, por Håkon Wium Lie, que veio a tornar-se co-criador de CSS. Tendo chegado outras propostas, estas foram discutidas publicamente, tendo grande contributo, Bert Bos, tornou-se co-autor CSS 1. Em 1996, considerada já completa a primeira versão, a W3C atribui recomendação a CSS 1 (CSS level 1) [W3C (c), 2015].

CSS2 (CSS Level 2), especificação desenvolvida e publicada como recomendação em 1998 pela W3C, tendo como funcionalidades as propriedades de posicionamento absoluto, relativo e fixo (*absolute, relative, and fixed*) dos elementos bem como *Z-index*, profundidade dos elementos. Com problemas de suporte e para corrigir certos erros, surgiu por parte da W3C, a versão CSS 2.1 (CSS revision 1). A primeira candidatura a recomendação surgiu em 2004, sendo recusada, para serem revistas certas alterações. Até 2010 foram feitas mais 3 candidaturas a recomendação e posteriormente desenvolvidas para correções e novas clarificações. Foi em 2011 publicada a versão 2.1 como sendo recomendação por parte da W3C. Quanto ao CSS3 (CSS level 3), teve e tem um percurso diferente das versões anteriores, sendo dividido em várias camadas. A que chamam de "Modules" (Módulos), devido essa separação, até ao momento foram publicados como principais recomendações os seguintes módulos: *color* (2011/06), *Selectors Level 3* (2011/09) e *Media Queries* (2012/9) [W3C (c), 2015].

Existem 3 formas de aplicar CSS, conforme se pode visualizar na imagem abaixo:

index.html	my.css
<pre>1 <!DOCTYPE html> 2 <html lang="pt"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <meta name="viewport" content="width=device-width, initial-scale=1.0"> 7 <meta name="description" content=""> 8 <meta name="author" content=""> 9 <title>Tese</title> 10 <link rel="stylesheet" href="my.css"> 11 <style> 12 body { 13 padding-top: 50px; 14 } 15 </style> 16 </head> 17 18 <body> 19 20 <header class="myClasse">cabeçalho </header> 21 22 <p style="color:blue">paragrafo</p> 23 24 <div >DIV</div> 25 26 <footer id="myID">Footer</footer> 27 </body> 28 </html></pre>	<pre>1 /* estilo a qualquer elemento p */ 2 div { 3 color: red; 4 font-size: 20px; 5 } 6 7 /* estilo a qualquer elemento com a myClasse */ 8 .myClasse { 9 color: blue; 10 font-size: 30px; 11 } 12 13 /* estilo a elemento com id */ 14 #myID { 15 color: green; 16 font-size: 18px; 17 }</pre>

Figura 17 - CSS: Aplicar CSS

- Diretamente no elemento html, como era feito inicialmente, aplicando a cada elemento as respectivas propriedades CSS, como se pode visualizar no elemento *p* que contém atributo *style*, é aplicado a cor azul;
- A segunda forma, ainda no documento *index.html*, é definição dos estilos dentro do elemento *style*, contido dentro em *head*, antes do elemento *body*, onde são definidos os elementos html, que no exemplo na figura é aplicado uma distância do topo do documento html de 50 pixels;
- A forma mais utilizada, até por ser a mais adequada, é aquela na qual todos os estilos são definidos num documento externo, quer estejam no mesmo servidor quer num á parte. Como foi referido anteriormente, esta possibilidade foi implementada a partir do HTML4. Após criar um ficheiro com extensão .css, conforme se pode visualizar na à direita, na figura 12, tem de se indicar os respetivo caminho do ficheiro, dentro do elemento head através do elemento html link, onde esse caminho é definido através do atributo href;
- Por último, é possível de forma mais dinâmica aplicar estilos através de JavaScript, que será demonstrado no próximo subcapítulo.

Para além de haver várias formas de aplicar os estilos definidos, existem também varias formas de indicar a que elemento(s) aplicar um determinado ou conjunto de estilos.

Esses procedimentos estão demonstradas na figura 12, mais à direita, da forma simples e direta:

- **Elemento:** indicando o nome do respetivo elemento, como exemplo na figura 12, está o elemento *a*, com o estilo atribuído de cor vermelho e tamanho de letra 20 pixéis, a todos os elementos *div* existentes no documento.
- **Classe:** para criar um estilo do tipo classe, quer no documento html no elemento *head*, dentro do elemento *style*, através de ponto (.) + [nome da classe], conforme se pode visualizar no lado direito da figura 12, foi criada a classe *.myClasse*, com respetiva formatação. Para aplicar a definição da classe, é preciso indicar no elemento que irá implementar aquela classe, através do atributo *class*, como se pode verificar no elemento *header*, onde a semelhança do anterior, todo elemento que contém aquela classe, será aplicada a respetivo estilo
- **ID:** ultima forma e a mais específica. Primeiro é necessário atribuir um identificador único, ou seja, dar um nome exclusivo ao elemento através do atributo *id*. Conforme se pode verificar na figura 12, é atribuído um *id* ao elemento *footer*, de *myID*, onde consequentemente é indicado que no ficheiro com extensão CSS, esse mesmo identificador é chamado através do caracter cardinal (#) + [nome do ID], ou seja *#myID*;
- Utilizar os procedimentos anteriormente referidos, ao mesmo tempo.

É de referir que as maneiras indicadas anteriormente são chamados de *selectors* (selectores), onde o primeiro é chamado de *Element Selector* (selector de elemento), a segunda forma de *Class Selector* (selector de classe) e por ultimo é chamado de *ID Selector* (selector de id)[MDN (b), 2015].

2.2.3.2 CSS3 – Evolução do CSS

Como mencionado anteriormente na história do CSS, o nível 3 do CSS teve uma abordagem mais aberta, ao estilo da especificação do HTML5 da WHATWG, como *Living Standard*, onde se vão desenvolvendo novas funcionalidades, conforme as necessidades através de pequenos módulos [Peter Gasston, 2015].

Para se seguir o mesmo tipo de raciocínio que autor Mathias Bynens fez para HTML5, transpondo o mesmo para o CSS3, dividindo em três grupos principais. Esta divisão é feita com base no livro de Peter Gasston, “The book of CSS3”, da última referência.

No grupo 1: *layout* e dispositivos. Enquadram-se os seguintes módulos: *media queries*, *multiple Columns*, *flexible box Layout*, *grid*, *layout* e *values / sizes* (que diz respeito a unidades de comprimento). Nesta divisão são agrupados todos os módulos relacionados de forma direta ou

indiretamente com a disposição dos elementos, ao longo de uma página web, transversalmente a todos os dispositivos existentes.

O grupo 2: formatação. Abrange os seguintes módulos: seletores, pseudo-classes (como: `link`, `:hover`, `:visited` e `:active`), pseudo-elementos (`::first-line`, `::before`, `::after`), fontes web, efeitos de texto e cor / opacidade. Estes agrupamentos dizem respeito à maneira que como se pode chegar a um ou a um conjunto de elementos, alterando as suas propriedades a nível de aspecto, mais concretamente no texto. Também de referir que o developer não foi limitado somente a fontes existentes localmente do computador do utilizador, podendo utilizar ou importar fontes armazenadas externamente através de `@font-face` ().

Finalmente, o grupo 3: manipulação e transformação. Incluem os seguintes módulos: *background image*, *border* e *box effects*, gradientes, transformação 2D, transformações 3D, transições, animações, *blend modes*, filtros, efeitos e máscara.

Neste último, reside a maioria das novidades do nível 3 do CSS, não menosprezando os anteriores, nomeadamente o primeiro que veio ajudar de forma significativa a maneira como interagimos com os mais diversos dispositivos e vai para além da estrutura e formatação de uma página web. Algumas funcionalidades podem ser alternativas ao JavaScript e outras um complemento ao JavaScript. Desde o tipo de transformação que se pode utilizar tanto em imagem como texto até ao nível de animação dos elementos existentes na página Web. Há algum tempo atrás, utilizava-se para criar interação ou maior dinâmica, o JavaScript e respetivas Frameworks ou bibliotecas, como é o caso de animações que, no CSS, já são possíveis através do módulo de animação e transição.

2.2.3.3 Vantagens & Desvantagens

Como na tecnologia anterior, o CSS tendo incorporado algumas da funcionalidade que se pode aplicar tanto através de JavaScript e algumas dessas novas funcionalidades serem baseadas na tecnologia SVG também se encontra pros e contras devido a quem é responsável pelo desenvolvimento dos Browsers.

Vantagens, com base no que foi dito anteriormente:

- Animações sem JavaScript Frameworks, bibliotecas ou *Plugins* externos;
- Maior facilidade na manipulação dos elementos, como arredondamento de imagens;
- Rápido carregamento das páginas HTML;
- Menos código JavaScript;
- Adaptação a qualquer dispositivo através, principalmente, do módulo *media queries*;
- Uso menor de recursos em termos de processamento, relativamente no Browser;
- Carregamento de fontes externas, que estão ou estejam num repositório Online.

Desvantagens:

- Para algumas funcionalidades é necessária ainda a utilização dos ditos prefixos: ms, webkit, moz. Devido a essas funcionalidades ainda estarem em fase de desenvolvimento;
- Ser o *Living Standard*, que se vai modificando ao longo do tempo;
- À semelhança da tecnologia HTML5, a maior dos Browser executar a funcionalidades de forma diferente.

2.2.4 JavaScript

A linguagem de programação JavaScript foi criado pela Netscape, mais precisamente por Brendan Eich, anteriormente conhecida pelos seguintes nomes: Mocha ou LiveScript. A sua primeira versão foi lançada em 1995, integrada numa nova versão Browser Netscape Navigator (versão 4), hoje em dia conhecido como Mozilla Firefox, suporte também a tecnologia Java, que antigamente ainda era detida pela Sun Microsystems e anos mais tarde pela Oracle e até aos dias de hoje [MDN (c), 2015].

A linguagem JavaScript foi enviada para ECMA, *European Computer Manufacturers Association* (Associação Europeia dos Fabricantes de Computadores), uma associação sem fins lucrativos, oficializou como nome padrão de ECMAScript, nome mais conhecido em eventos dedicados a programadores, como por exemplo a Google I/O [Rauschmayer, 2014].

Esta linguagem de programação ou script, tem como objetivo principal acesso a documentos, mais precisamente, a elementos de um documento HTML, através do DOM (Document Object Model). Permite, por exemplo, criar, apagar, esconder e animar elementos. Com a chegada do HTML5 e a respetivas API, constitui a espinha dorsal para os Browsers modernos. Ou seja, ainda mais fundamental para as páginas serem dinâmicas e mais interativas, para funcionalidades como geolocalização e Canvas. [Robin Nixon, 2015]. Até meio deste ano (2015), o *Standard* desta linguagem encontrava-se na versão 5.1, ou seja, ECMAScript 5.1. Em junho deste ano, foi oficialmente aprovado versão ECMAScript 2015 (6ª Edição), normalmente referida como ES6. Que se aproximará das linguagens mais tradicionais e conhecidas, como Java e C# (Sharp) e com suporte total em tudo os Browsers até ao final do ano de 2016 [Mozilla, 2015].

À semelhança da tecnologia CSS, existem também várias formas de incorporar ou chamar código JavaScript num documento HTML, conforme se pode visualizar na figura 13, abaixo:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>JS</title>
6
7   <!-- indicação do caminho do ficheiro externo-->
8   <script src="externo1.js"></script>
9
10  <script>
11    function myFunction() {
12
13      /*preenchimento do elemento p através da sua classe de forma mais tradicional do JS*/
14
15      document.getElementsByClassName("demo")[0].innerHTML = "Paragraph changed.";
16
17      /* alteração do estilo de um dos botões de forma tradicional do JS através do ID*/
18      document.getElementById("btn").style.fontSize = "30px";
19    }
20  </script>
21 </head>
22
23 <body>
24
25   <p class="demo">A Paragraph</p>
26
27   <a class="myClass" href="#" onclick="myFunction()">Try it</a>
28
29   <a id="btn" href="#" >Try it 2</a>
30
31
32   <script src="externo2.js"></script>
33
34
35   <script>
36
37     /* associação JS entre elemento e a função com respetivo evento, através de selectores ( querySelector) */
38
39     document.querySelector("#btn").addEventListener("click", myFunction2);
40
41     function myFunction2() {
42       alert("Hello World!");
43
44       /* alteração da cor do botão através de JS e com o selectores */
45       document.querySelector("#btn").style.color = "red";
46     }
47   </script>
48 </body>
49 </html>
```

Figura 18 - JavaScript: Incorporar

O exemplo acima, ainda na versão 5.1 por falta de suporte nos Browsers, é baseado em exemplo do W3Schools, com algumas alterações, mais especificamente na utilização de *querySelector*.

Relativamente às formas de incorporação do JavaScript, existem duas:

- No documento HTML: existem duas possibilidades, sendo a primeira, colocar no início do documento, mais concretamente no elemento head, dentro elemento *<script>*, conforme se pode visualizar na figura acima, onde também demonstra que existe a hipótese de se colocar no fim do documento mais concretamente antes de se fechar o elemento *body*, também dentro de elemento *<script>* .
- Ficheiro Externo: à semelhança do CSS, é também possível indicar o respetivo caminho dentro o elemento *<script>* através do atributo *src* , onde este ficheiro pode estar guardado no servido local ou num servido externo, indicando o respetivo url;
- Utilizar a duas formas anteriormente referidas, ao mesmo tempo.

2.2.5 Bibliotecas e Frameworks

Nesta secção são apresentadas Frameworks ou bibliotecas destinadas tanto para desenvolvimento de interfaces gráficas bem como associadas às tecnologias anteriormente descritas, para um melhor e eficaz desenvolvimento das funcionalidades propostas, a incluir no projeto.

2.2.5.1 jQuery

jQuery é biblioteca JavaScript, cujo objetivo é simplificar a maneira como se escreve aplicações em JavaScript. Desde logo facilita o acesso aos elementos / tags HTML que constituem o DOM, modificando as suas propriedades através dos métodos existentes em JavaScript, quer através dos atributos dos elementos HTML. Esta biblioteca contém funções que possibilitam o rápido acesso ao DOM e outras funções, que além facilitar a compreensão de quem está a desenvolver, permite fazer mais com muito menos linha de código. Uma das funcionalidades mais utilizadas, é esconder ou mostrar um determinado elemento através certo evento, ou criar animações como *fade-in* ou *fade-out*, ou até mesmo alterar o tamanho do elemento quer através propriedades JavaScript ou dos atributos CSS [Brad Dayley, 2013].

Podem visualizar alguns dos exemplo mencionados acima, na figura seguinte:

```
1 <!doctype html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>jQuery</title>
7   <style>
8     div {display: none;}
9   </style>
10 </head>
11 <body>
12   <button id="btn">Change</button>
13   <p class="demo">paragrafo</p>
14
15   <button id="btn2">Change 2</button>
16
17   <div>
18     <p>01a</p>
19     <p>01a2</p>
20   </div>
21   <!-- definição do caminho da framework JS -->
22   <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
23   <script>
24     /* aplica inicialmente a cor verde ao botão2 */
25     $("#btn2").css("color", "green");
26     /* evento com primeiro botão */
27     $("#btn").click(function (event) {
28
29       event.preventDefault();
30       /*Aplicar formatações com jQuery*/
31
32       $(".demo").css("font-size", "40px");
33
34       $(this).hide();
35       $(".demo").show();
36     });
37
38     $("#btn2").on("click", function () {
39       $("#btn").show();
40
41       $(".demo").css("font-size", "12px");
42
43       $(".demo").hide();
44     });
45   </script>
46 </!
47 </!>
```

Figura 19 - jQuery: Exemplos

Na figura 19, são demonstrados alguns das capacidades desta Frameworks, como por exemplo esconder e mostrar certos elementos identificados quer por um determinado id ou classe e atribuir os estilos CSS a alguns elementos como mudar cor ou tamanho de cor.

Para aceder a um elemento através de um id, colocar como se encontra na figura 19, `$("#btn")`, que seleciona um elemento com id "btn" através do símbolo # (cardinal) ou para aceder a uma classe CSS, como por exemplo classe demo através `$(".demo")`, com ponto (.), antes do nome da classe. Tudo à semelhança da função JavaScript *querySelector*, demonstrado no subcapítulo JavaScript.

Ao longo dos anos esta biblioteca foi ganhando grande destaque, onde os responsáveis resolveram criar mais outras duas bibliotecas complementares: jQuery UI e jQuery Mobile, com finalidades diferentes.

jQuery UI faculta efeitos, interações, *widgets* e temas para acelerar a incorporação / criação de elementos mais complexos de forma mais eficiente, que irão constituir na nossa aplicação ou Website [Brad Dayley (b), 2013].

Por outro lado jQuery Mobile, devido ao aparecimento *Smartphones*, houve a necessidade de criar Website ou aplicações mais acessíveis a este tipo de dispositivos, bem como a quem está desenvolver para este mesmo dispositivo. Esta biblioteca destaca-se para além da inclusão de simples componentes / elementos UI, com as respetivas interações destinadas a esta plataforma, destaca-se pelo facto de camuflar o grau de complexidade de adaptação dos elementos que constituem quer o Website ou aplicação [Brad Dayley (c), 2013].

2.2.5.2 CamanJS

CamanJS, *Canvas Manipulation*, framework de JavaScript providencia funcionalidades de edição de imagem de forma simples e eficiente, quer de forma direta através do elemento Canvas, quer indiretamente através do elemento `` [JStier, 2015].

A funcionalidade principal abrange existência de uma vasta lista tanto de filtros como de efeitos.

Na figura abaixo, encontra-se uma demonstração da framework, à qual são aplicados um filtro e um efeito a uma determinada imagem.

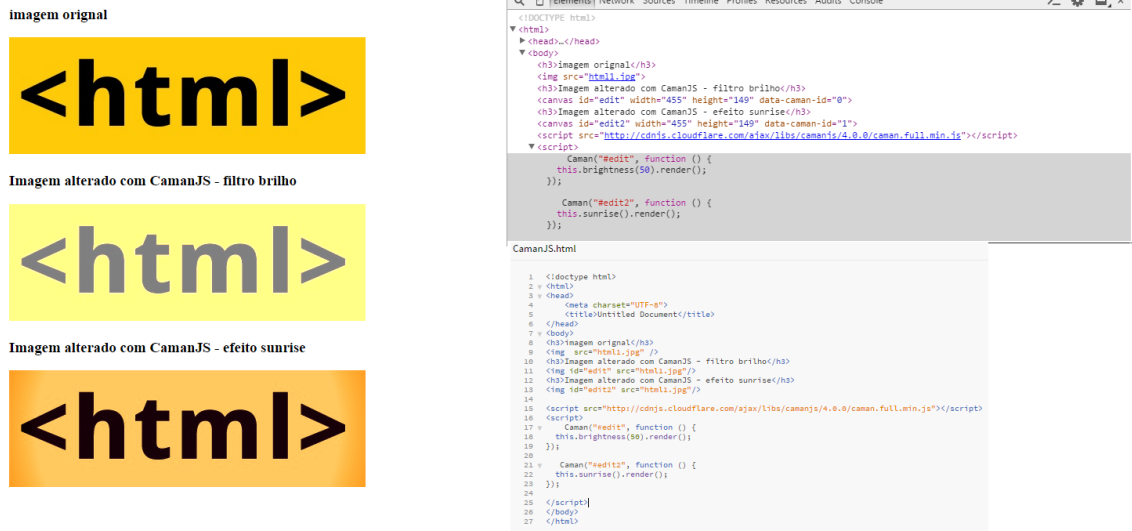


Figura 20- CamanJS: Demonstração

Na figura 15 é possível verificar no lado direito e no fundo da imagem, o respetivo código dos elementos com nomes /ids edit e edit2. Foram criados como elementos imagem, mas ao aplicar as filtros ou efeitos, por exemplo, a Frameworks converte o elemento imagem para Canvas e executa as respetivas funções, o filtro de brilho a 50% e o efeito Sunrise, como se pode verificar no lado direito da figura.

Também é possível criar novos efeitos através da funcionalidade Pixel-wise Filters, modificando os pixéis através de valores RGBA (Red, Green, Blue, Alfa).

Por último há opção de fazer modificação da imagem através o Kernel de Convulsão, que é a matriz de picheis que constitui a imagem [CamanJS, 2015].

2.2.5.3 FabricJS

É uma biblioteca JavaScript que se destina ao elemento Canvas, permite facilitar desenhar de formas geométricas.

Segundo o website oficial permite criar e popular objetos sobre o elemento Canvas, como por exemplo formas geométricas simples tais como retângulos, círculos, elipses, polígonos e entre outras formas mais complexas que são constituídas por centenas ou milhares de simples *Path* (caminhos) [Mahesh Chand, 2012].

Admite operações como rotação, escalar com o rato, bem como modificar as propriedades dos objetos como transparência, cor e profundidade (z-index). Também está disponível a funcionalidade de agrupar os objetos.

Contém uma pequena lista de filtros que se podem aplicar em imagens, que vai *desde grayscale, brightness, invert, mask, sepia*, etc [FabricJS, 2015]. Alguns dos filtros anteriormente mencionados, pode-se ver a sua execução na seguinte imagem:

Imagem Original



Imagem com Filtro de cinza



aplicação de Múltiplos Filtros



```

1 <!doctype html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>FabricJS</title>
7 </head>
8
9 <body>
10 <h3>Imagem Original</h3>
11 
12 <h3>Imagem com Filtro de cinza</h3>
13 <canvas width="500" height="150" id="c"></canvas>
14 <h3>aplicação de Múltiplos Filtros</h3>
15 <canvas width="500" height="150" id="cd"></canvas>
16 <!--Carregamento da framework-->
17 <script src="http://cdnjs.cloudflare.com/ajax/libs/fabric.js/1.5.0/fabric.min.js"></script>
18 <script>
19   var canvas = new fabric.Canvas('c');
20
21   fabric.Image.fromURL('html1.jpg', function (img) {
22
23     // add filter
24     img.filters.push(new fabric.Image.Filters.Grayscale());
25
26     // apply filters and re-render canvas when done
27     img.applyFilters(canvas.renderAll.bind(canvas));
28
29     // add image onto canvas
30     canvas.add(img);
31   });
32
33   var canvas2 = new fabric.Canvas('cd');
34   fabric.Image.fromURL('html1.jpg', function (img2) {
35
36     img2.filters.push(
37
38       // aplica o filtro de Sepia
39       new fabric.Image.Filters.Sepia(),
40       // aplica o filtro de brilho
41       new fabric.Image.Filters.Brightness({
42         brightness: 90
43       })
44     );
45     // aplica o filtro de ruído
46     new fabric.Image.Filters.Noise({
47       noise: 90
48     });
49   });
50   // atualiza a alterações
51   img2.applyFilters(canvas2.renderAll.bind(canvas2));
52
53   // mostra as modificações no canvas
54   canvas2.add(img2);
55
56 </script>
57 </body>
58 </html>

```

Figura 21 - FabricJS: Demonstração

Na figura 21, para além de se poder visualizar a imagem original, a segunda imagem demonstra o resultado do filtro de *grayscale*. A sua implementação pode verificar-se no lado direito da figura 21.

Na última imagem, cujo código se encontra no lado direito da figura, são executados três filtros: sépia, brilho e de ruído, sendo que estes dois últimos recebem um argumento com a quantidade aplicar, mais precisamente de 90.

Finalmente, transforma todos objetos existentes no elemento Canvas no formato de dados Json, bem como exportar para SVG. [FabricJS (b), 2015]

2.2.5.4 CreateJS

Tal como o Website oficial, é uma *Suite* com constituída por 4 bibliotecas modulares: EaselJS, TweenJS, SoundJS e PreloadJS.[CreateJS, 2015]

A biblioteca mais relevante, no que toca este documento, é EaselJS que está relacionada com a manipulação e interatividade dentro do elemento Canvas. A API é baseada ou semelhante a Adobe Flash / ActionScript 3.0, no qual os objetos estão alinhados na lista hierárquica.

Destinava-se sobre o desenvolvimento de jogos sobre elemento Canvas [Manderscheid, 2014].

Imagem Original



Imagem com Filtro de cinza



Imagem com Filtro de HUE / ALTERAÇÃO DE COR



```
1 <!doctype html>
2 <html>
3
4 <head> ... </head>
5
6
7
8
9 <body onload="init();">
10 <h3>Imagem Original</h3>
11 
12 <h3>Imagem com Filtro de cinza</h3>
13 <canvas width="500" height="150" id="c"></canvas>
14 <h3>Imagem com Filtro de HUE / ALTERAÇÃO DE COR</h3>
15 <canvas width="500" height="150" id="cd"></canvas>
16 <!--Carregamento da framework-->
17 <script src="https://code.createjs.com/easeljs-0.8.1.min.js"></script>
18 </script>
19
20 var img;
21 function init() {
22     //wait for the image to load
23     img = new Image();
24     img.onload = handleImageLoad;
25     img.src = "html1.jpg";
26 }
27 function handleImageLoad() {
28     //Find canvas and load images, wait for last image to load
29     var canvas = document.getElementById("c");
30     // cria nova tela
31     stage = new createjs.Stage(canvas);
32     var bmp = new createjs.Bitmap(img);
33     stage.addChild(bmp);
34     var blurFilter = new createjs.BlurFilter(20, 0, 1);
35     bmp.filters = [blurFilter];
36     bmp.cache(0, 0, img.width, img.height);
37     stage.addChild(bmp);
38     stage.update();
39     //criação da segunda tela
40     var canvas2 = document.getElementById("cd"),
41     stage2 = new createjs.Stage(canvas2),
42     bmp2 = new createjs.Bitmap(img);
43     stage2.addChild(bmp);
44     var matrix = new createjs.ColorMatrix().adjustHue(80);
45     bmp2.filters = [new createjs.ColorMatrixFilter(matrix)];
46     bmp2.cache(0, 0, img.width, img.height);
47     stage2.addChild(bmp2);
48     // draw to the canvas:
49     stage2.update();
50 }
51 </script>
52
53 </html>
```

Figura 22 - EaselJS: Demonstração

A figura acima demonstra a implementação de dois filtros, uma em cada imagem. Na segunda imagem é aplicado o *blur* (desfocar) e na terceira imagem é aplicado o filtro de alteração de cor, *HUE*, que varia entre os valores de cores de 0 a 360.

A segunda, TweenJS, é uma poderosa ferramenta que permite animar, de forma simples e eficiente, os objectos criados através da biblioteca anterior [Manderscheid (b), 2014].

A penúltima desta *Suite*, é SoundJs, como próprio nome indica, destina a permite adicionar, associar ou aplicar após ou durante certo eventos um executar ficheiro de som, onde com esta biblioteca evita que haja ciclos infinitos e previne ações indesejadas com ficheiros de som como delas [Manderscheid (c), 2014].

Por último, faz parte desta *Suite* é PreloadJs, onde esta ferramenta cujo objetivo é carregar todos os recursos a serem usados num jogo ou aplicação [Manderscheid (d), 2014].

2.2.5.5 HammerJS

É uma biblioteca open-source dedicada ao reconhecimento de gestos por Touch, rato, etc [HammerJS, 2015].

Permite a evocação de múltiplas instâncias / eventos ao mesmo tempo, na criação de interfaces interativas. Cujo foco principal são os dispositivos móveis ou dispositivos de superfícies que permitem a interação quer através os dedos quer com uma caneca, como quadros interativos [Eliot Sykes, 2014].

Na figura abaixo encontra-se um exemplo com a funcionalidade ou movimento *Swipe*:



Figura 23 - HammerJS: Demonstração

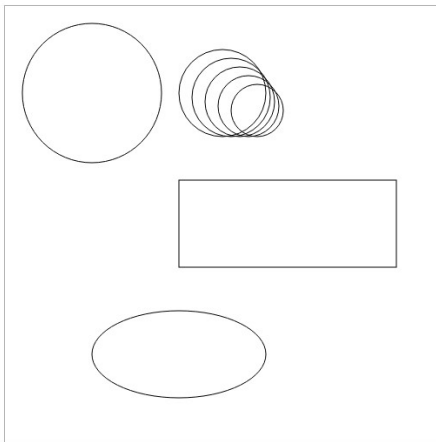
2.2.5.6 Raphaël

É uma pequena biblioteca JavaScript que tem o propósito de simplificar o trabalho com gráficos vetoriais na web.

Utilizada SVG para a criação de gráficos e os consequentes objetos a serem inseridos no DOM, onde posteriormente permite manipular os elementos com as funções JavaScript fornecidas, como animações e transformações. [James Williams, 2011]

Na figura abaixo é demonstrado a facilidade como se pode criar formas geométrica através desta biblioteca.

Raphael SVG Exemplos



```
1 <!doctype html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>CreateJS-eASEjs</title>
7   <style type="text/css">
8     #tela {
9       border: 1px solid #aaa;
10    }
11  </style>
12 </head>
13
14 <body>
15   <h3>Raphael Exemplos http://raphaeljs.com </h3>
16   <svg id="tela"></svg>
17   <!-- carregamento da framework -->
18   <script src="raphael-min.js"></script>
19   <script>
20     window.onload = function () {
21       var paper = new Raphael(document.querySelector('#tela'), 500, 500);
22       var circle = paper.circle(100, 100, 80);
23       for (var i = 0; i < 5; i += 1) {
24         var multiplier = i * 5;
25         paper.circle(250 + (2 * multiplier), 100 + multiplier, 50 - multiplier);
26       }
27       var rectangle = paper.rect(200, 200, 250, 100);
28       var ellipse = paper.ellipse(200, 400, 100, 50);
29     }
30   </script>
31 </body>
32 </html>
```

Figura 24 - Raphael: Demonstração

Na figura acima, é feita a demonstração de como desenhar formas básicas desde círculos a elipses com poucas linhas de código.

É de salientar a compatibilidade com os Browser mais antigos como desde o Firefox 3.0, Safari 3.0, Google Chrome 5.0, Opera 9.5 e Internet Explorer 6.0. [Raphael, 2015]

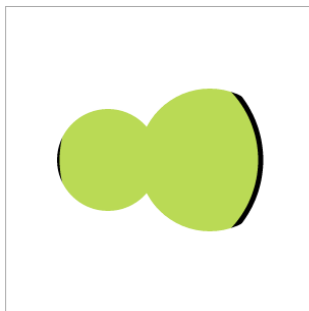
2.2.5.7 Snap.SVG

É uma recente biblioteca de JavaScript, como a anterior, destinada para tecnologia SVG, suportada pela Adobe, criada pelo mesmo autor de Raphaël, contrariamente à anterior que era direcionada a dar compatibilidade aos Browsers mais antigos. Esta nova biblioteca vira-se mais para os Browsers mais recente, usufruindo assim da especificação mais recentes, usufruindo assim da especificação mais recente desta tecnologia.

Para além das operações básicas de SVG como criação de forma geométrica, animações e transformações, permite aplicar padrões (*patterns*), gradientes mais complexos, agrupar e aplicar máscaras. [SnapSvg, 2015].

Na figura abaixo, encontra-se um pequeno exemplo a desenhar dois círculos e animar um deles após o carregamento página.

Snap SVG Exemplos by <http://snapsvg.io>



```
1 <!doctype html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Snap</title>
7   <style type="text/css">
8     #tela {
9       width: 500px;
10      border: 1px solid #aaa;
11    }
12  </style>
13 </head>
14
15 <body>
16
17
18   <h3>SNAP SVG Exemplo by http://snapsvg.io</h3>
19   <svg id="tela"></svg>
20   <!--Carregamento da framework-->
21   <script src="snap.svg-min.js"></script>
22   <script>
23     // First lets create our drawing surface out of existing SVG element
24     // If you want to create new surface just provide dimensions
25     // like s = Snap(500, 500);
26     var s = Snap("#tela");
27     // Lets create big circle in the middle:
28     var bigCircle = s.circle(150, 150, 100);
29     // By default its black, lets change its attributes
30     bigCircle.attr({
31       fill: "#bada55",
32       stroke: "#000",
33       strokeWidth: 5
34     });
35     // Now lets create another small circle:
36     var smallCircle = s.circle(100, 150, 70);
37     // Lets put this small circle and another one into a group:
38     var discs = s.group(smallCircle, s.circle(200, 150, 70));
39     // Now we can change attributes for the whole group
40     discs.attr({
41       fill: "#fff"
42     });
43     // Now more interesting stuff
44     // Lets assign this group as a mask for our big circle
45     bigCircle.attr({
46       mask: discs
47     });
48     // Despite our small circle now is a part of a group
49     // and a part of a mask we could still access it:
50     smallCircle.animate({r: 50}, 1000);
51   </script>
52 </body>
53
54 </html>
```

Figura 25 - Snap SVG: Demonstração

2.2.5.8 Zurb Foundation

Zurb Foundation (<http://foundation.zurb.com>) é uma das Frameworks mais populares no que toca ao desenvolvimento de interfaces gráficas responsivas [Patel, 2014]. Atualmente encontra-se na versão 5, onde segue a linha de orientação de desenvolvimento de “*mobile first*”, ou seja, *design* do website ou aplicação a pensar primeiro em dispositivos móveis e posteriormente alargar para os restantes dispositivos de grandes dimensões.

Um dos mais benefícios desta Framework, é a utilização de unidades de medidas mais recentes, como `rem`, que é a unidade que permite adapta de forma correta o tamanho do texto a qualquer dispositivo.

À semelhança de Frameworks do género, tem o propósito de agilizar o desenvolvimento de interfaces gráficas, capazes a funcionar de forma correta em qualquer dispositivo. Para isso são desenvolvidos vários componentes através de tecnologias mais recentes, como CSS3, JavaScript e recentes funcionalidades do HTML5.

Um dos componentes, a título de exemplo, é o chamado *Off-canvas*. Ao aplicar este componente, nos dispositivos móveis, o menu fica inicialmente escondido, bastando pressionar num ícone ou botão para este aparecer. Com isto a Website ou aplicação terá maior espaço de utilização, conforme se pode visualizar na figura abaixo:

The image displays two side-by-side panels. The left panel, titled 'HTML', shows the source code for an off-canvas menu. It includes a toggle link with the class 'left-off-canvas-toggle', a hidden menu with the class 'left-off-canvas-menu' containing a list of items, and an exit link with the class 'exit-off-canvas'. The right panel, titled 'RENDERED HTML', shows the visual output. It features a 'Menu' section with a list of items (Item 1 through Item 7) on the left side of the page, and the main content area on the right. A red text label 'Depois de Pressionar em Menu' is positioned between the two rendered HTML sections, indicating the state after the menu is toggled.

Figura 26 - Zurb Foundation: Código Exemplo

Esta Framework, contém componentes de estruturação como grid (grelha), para uma elaboração do *Layout* do Website ou aplicação, com um aspeto visual correto, qualquer que seja o dispositivo. Também componentes de navegação, como o exemplo anterior, bem como componentes direcionados a formulários e outros elementos multimédia tais como imagens e vídeos.

2.2.5.9 Twitter Bootstrap

Esta framework Twitter Bootstrap (<http://getbootstrap.com>), à semelhança da anterior, esta framework é uma coleção de ferramentas que tem o propósito de ajudar programadores a criar Websites responsivos. Onde hoje ainda existem Websites que não são responsivos ou adaptáveis, ou seja, para uma correta visualização só podem ser acedidos através de desktops, excluído outros dispositivos de dimensões mais pequenas como *Tablet* ou *Smartphones* que não podem usufruir corretamente do Website ou aplicação.

Com o objetivo de desenvolver Websites mais acessíveis aos mais diversos dispositivos, Jacob Thornton e Mark Otto criaram esta ferramenta através de JavaScript, mais concretamente através de jQuery e com classes CSS predefinidas a serem aplicadas nos elementos HTML [Kevin Lyn, 2015]

Através das tecnologias, CSS e JavaScript, é possível criar componentes, baseados HTML, como *Slideshow*, *Menus*, *tabelas*, *pop-up*.

Selecionando o Componentes semelhante à da framework anterior, na figura abaixo, pode visualizar-se uma proposta diferente de aparecimento do menu em dispositivos móveis, de forma mais dinâmica e que também são escondidos, aparecendo no topo e não de lado, como a anterior.

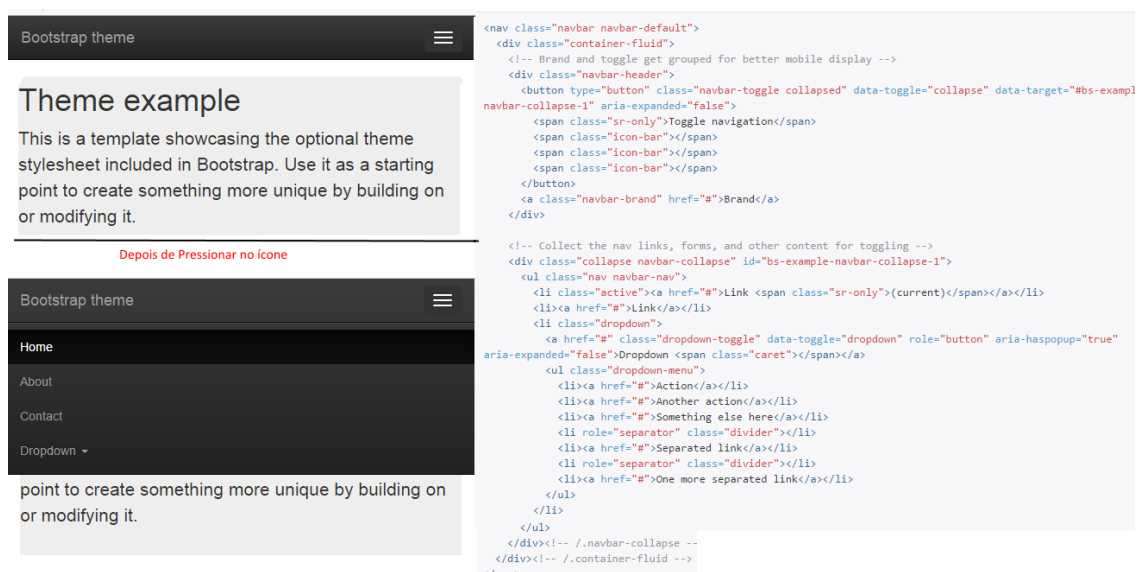


Figura 27 - Twitter Bootstrap: Exemplo Código

Atualmente encontra-se na versão 3.3.5, sendo que a versão 4 já se encontra em fase Alfa. À semelhança da framework anterior, existem vários componentes quer através CSS, quer através de JavaScript. Destacam-se o Modal, ao estilo de um pop-up, o Carousel que tem o objetivo de funcionar como *Banner* ou *spot* publicitário.

2.2.6 Conclusão

Para além do estudo e análise aos programas de edição e manipulação, existentes, são apresentadas neste capítulo possíveis tecnologias a serem selecionadas para o desenvolvimento da aplicação web.

A tecnologia da Adobe Flash, por omissão será excluída, embora tenha sido apresentada por ser a tecnologia mais usada nos programas anteriormente analisados, tendo também servido para contextualizar a sua utilização nos dias de hoje.

As tecnologias bem como frameworks ou bibliotecas aqui demonstradas servem para provar que existem duas ou mais tecnologias web que são alternativas válidas à tecnologia Adobe Flash.

Um dos objetivos deste documento é demonstrar através da nova geração de tecnologias web, é possível desenvolver Software tão bom ou melhor do que com tecnologia Adobe Flash. Sem auxílio de qualquer *Plugin*, bem como capaz de se adaptar a qualquer dispositivo quer fixo como móvel.

Somente algumas das tecnologias e frameworks ou bibliotecas serão utilizadas. Após a seleção de uma das tecnologias que estão dentro da especificação HTML5 - SVG e Canvas – será feita a seleção de Frameworks ou bibliotecas aqui apresentadas e direcionadas a cada tecnologia.

Para interface gráfica, para além da nova especificação do CSS, mais precisamente CSS3, irá ser escolhida a Framework para acelerar o desenvolvimento da interface gráfica: Zurb Foundation ou Twitter Bootstrap.

Por fim, é de salientar que, quer com base no estudo de soluções de edição e manipulação de imagem, quer através de algumas tecnologias web aqui referidas, será desenvolvido um protótipo de um editor de imagem Web-Based, cujo respetivo processamento será feito no lado do cliente, ou seja, no Browser (navegador) do utilizador.

3 Projeto

Este capítulo está dividido em quatro grandes secções: Análise do Sistema, Protótipo, Implementação e Conclusão.

A primeira secção diz respeito à descrição dos elementos de *hardware*, de software, à forma como eles comunicam com o utilizador e descrições de possíveis ações do utilizador com o projeto proposto. É também apresentado um conjunto de funcionalidades a introduzir no sistema, divididas em duas vertentes, de obrigatórias e complementares e rascunhos da possível interface gráfica e respetiva organização. Por último é apresentado o Website do projeto, é se pode encontrar, a maior parte dos conteúdos apresentados neste documento.

Na segunda secção, é demonstrado e descrito o protótipo do projeto proposto, explicando a sua navegação, as tecnologias utilizadas (daquelas apresentadas no capítulo estado da Arte), bem como o software para o seu desenvolvimento. É feita uma breve descrição dos módulos implementados, indicando as funcionalidades que constituem e uma apresentação de algumas imagens que ilustram o protótipo. Seguidamente foi feita a avaliação do protótipo através de inquéritos disponibilizados algumas pessoas que o testaram.

A terceira secção, foca de forma mais técnica, cada módulo, recorrendo de imagens com código, indicadas as dificuldades encontradas e possíveis formas de melhorar cada módulo e protótipo em geral.

Por último, conclusão são retiradas as ilações deste capítulo.

3.1 Análise do Sistema

Este primeiro subcapítulo, refere-se a uma análise funcional do projeto, englobando elementos que consistem o ecossistema e as funcionalidades do sistema proposto.

3.1.1 Arquitetura

Nesta secção são apresentados os meios evolutivos aquando da execução da aplicação.

Esses componentes que envolvem o ecossistema, relevante para o projeto, na figura abaixo. Onde posteriormente é feita uma descrição de cada componente presente na referida figura.

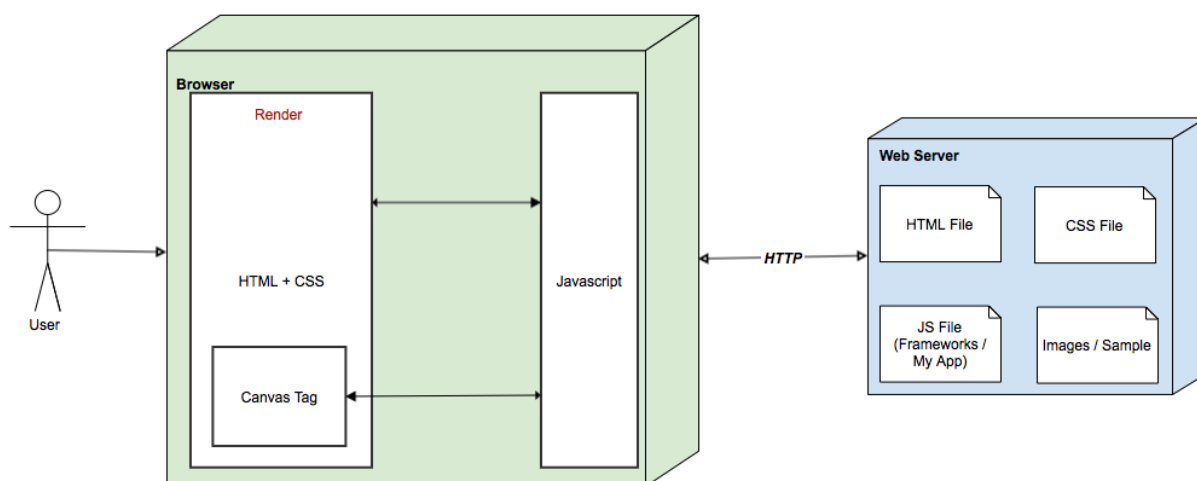


Figura 28 - Arquitetura do Sistema Proposto

De acordo com a figura acima, o ecossistema está dividido em duas partes, cliente (Browser) e Servidor.

Onde cliente tem o propósito:

- Ser o intermediário entre o utilizador e aplicação, que está armazenada no servidor;
- Fazer e receber pedidos de / para ao servidor conforme ação proferida pelo utilizador;
- Ser o programa responsável pelo *render* (processamento) dos elementos HTML existentes no ficheiro HTML bem como estilos existentes no ficheiro CSS.
- Contém o motor de JavaScript, responsável pela execução da linguagem JavaScript. E tratar de eventos mais interativos entre utilizador e aplicação, definida num ficheiro JavaScript.

Por outro lado o servidor tem o propósito:

- Armazenar a aplicação e respetivos recursos, como ficheiros e imagens;
- Responder aos pedidos feitos no lado do cliente;
- Transferir os ficheiros necessários para correr aplicação;
- Transferir a imagem exemplo caso seja pedido pelo utilizador.

3.1.2 Funcionalidades

Para o desenvolvimento do projeto foram traçadas quais as funcionalidades mais relevantes bem como selecionadas aquelas que iriam acrescentar valor ao protótipo.

É de referir que esta seleção foi feita com base das soluções exploradas, apresentadas no capítulo estado da Arte.

Como funcionalidades mínimas são propostas:

- *Upload* (carregamento) de Imagem (Disco)
- *Upload* (carregamento) de imagem Sample (do servidor)
- Guardar imagem para o disco do utilizador
- Rotação
- *Flip* vertical e Horizontal
- Filtros
- Efeitos
- *Crop*

Como opcionais ou complementares, poderão acrescentar valor aplicação:

- Captura de imagem através da Webcam
- Carregamento de imagem por URL
- *Zoom* (in/out)
- Texto
- *Layout Responsive*
- *Pan* (movimentar imagem ao longo do browser)
- Molduras
- Desenho Vectorial (desenho de figuras geométricas)
- Alteração de cores / *Theme* da Gui
- Pallette de cores
- Criação de *Url* da imagem para o utilizador
- *Fullscreen mode*

3.1.3 Casos de Uso

Nesta seção são abordadas todas as situações com as quais o utilizador pode-se deparar. Elas são descritas na imagem abaixo e explicadas posteriormente.

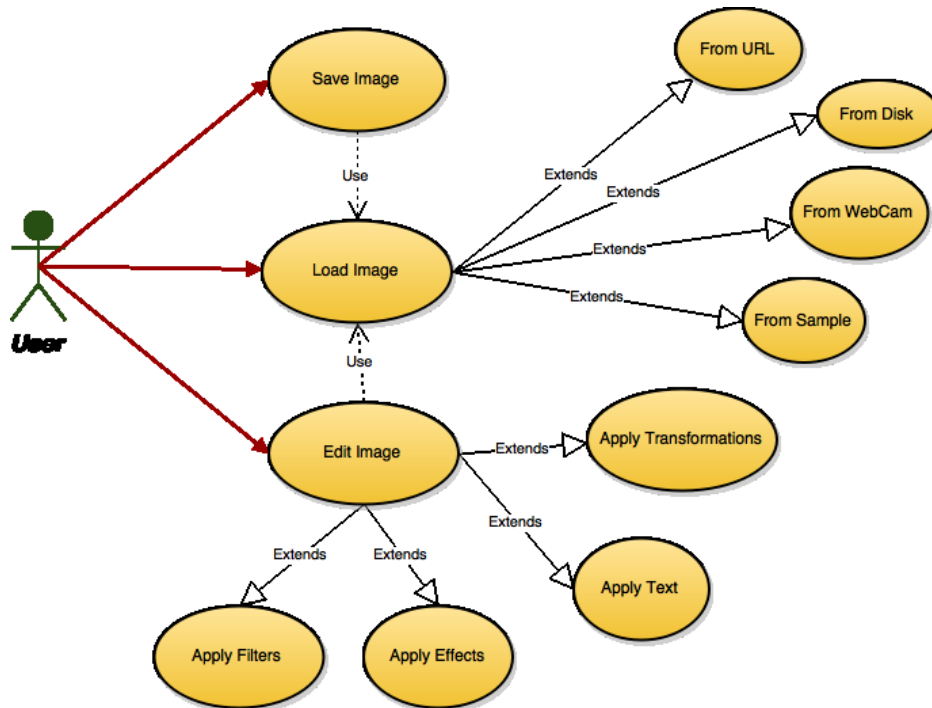


Figura 29 - Casos de Uso do sistema

Descrição da figura anterior:

- “User” (Utilizador): visto não haver restrição de login, qualquer utilizador pode executar qualquer operação sem limitações;
- “Load Image”: primeira ação que o utilizador deve fazer para poder aplicar as várias operações disponíveis, podendo ser executada de 4 formas, a serem descritas de seguida;
- “From disk”: operação responsável por fazer *upload* / carregar a imagem vindo do disco rígido do utilizador;
- “From Webcam”: é dada a possibilidade de utilizador, caso tenha disponibilidade, de capturar imagem através de uma Webcam do dispositivo em questão;
- “From URL”: o utilizador, poderá fornecer um URL de uma imagem;
- “From Sample”: possibilita ao utilizador carregar uma imagem disponível no servidor da aplicação;
- “Apply Transformations”: contém as funcionalidades elementares aplicadas à imagem, como rotação, Flip vertical, etc.
- “Apply Filtres”: ação responsável após seleção de um dos filtros disponíveis, será aplicado o filtro desejado
- “Apply Effects”: à semelhança do anterior, após a seleção de um dos efeitos, o utilizador tem a possibilidade de aplicar após pré-visualização do mesmo.
- “Apply Text”: possibilidade de inserção de texto bem como a respetivas propriedades
- “Save Image”: Após da execução de uma das ações poderá guardar a respetiva imagem com o resultado final desejado.

3.1.4 MockUps

Nesta secção podem visualizar-se as primeiras ideias e/ou rascunhos do *layout* da aplicação e como está previsto a sua organização. A organização e ideias aqui demonstradas, estão baseadas nos programas estudados no capítulo Estado da Arte.

Na figura abaixo ilustra a ideia inicial como estará organizada a aplicação: menu para funcionalidades como filtros, efeitos e ferramentas de correção rápida.

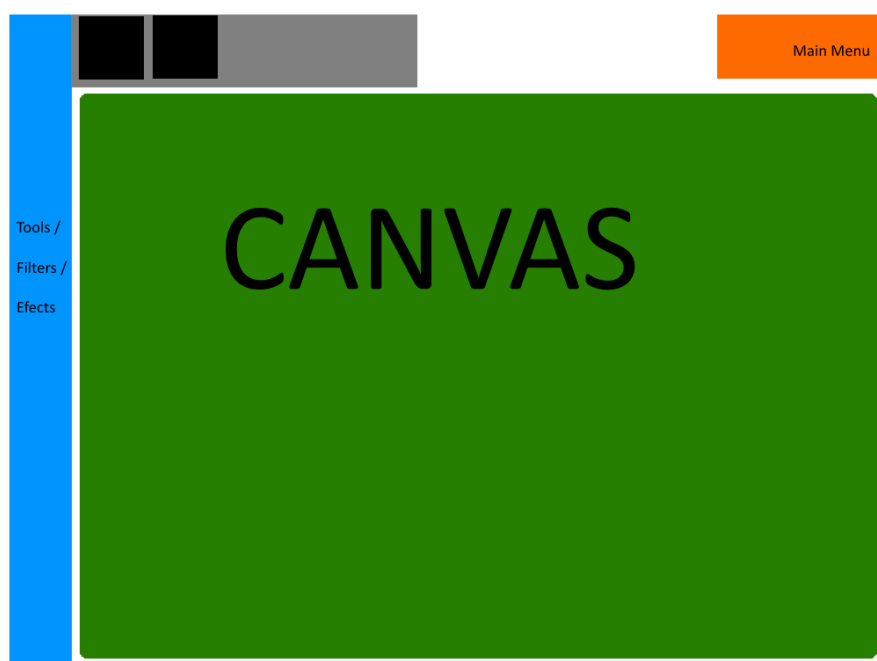


Figura 30 - MockUp de Organização

Na próxima figura abaixo, ilustra um rascunho do layout, cujo topo encontra-se funcionalidades gerais, enquanto no lado direito as ferramentas dedicadas à manipulação da imagem.

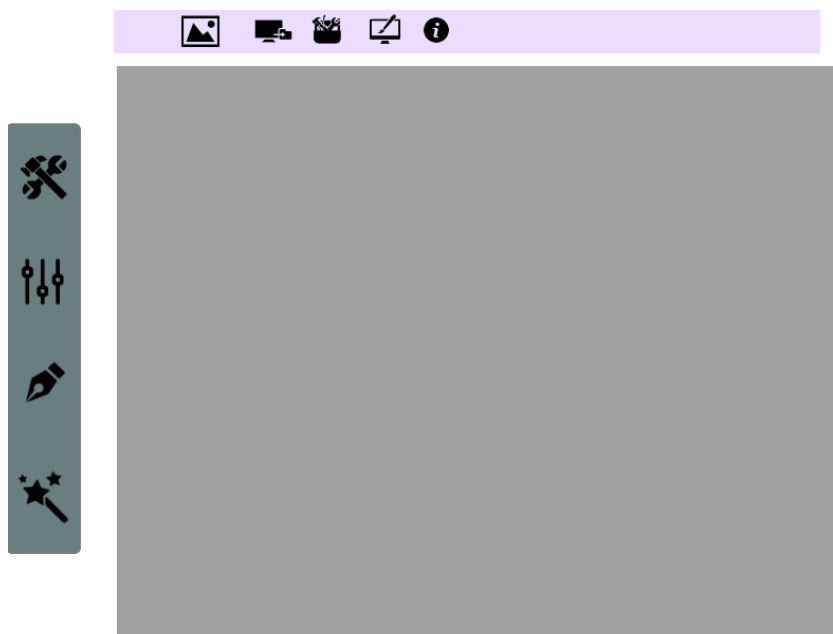


Figura 31 - Mockup do Layout da Aplicação

Por último, aparece o *layout* como apareceria caso seja seleciona uma opção da barra do lado esquerdo.

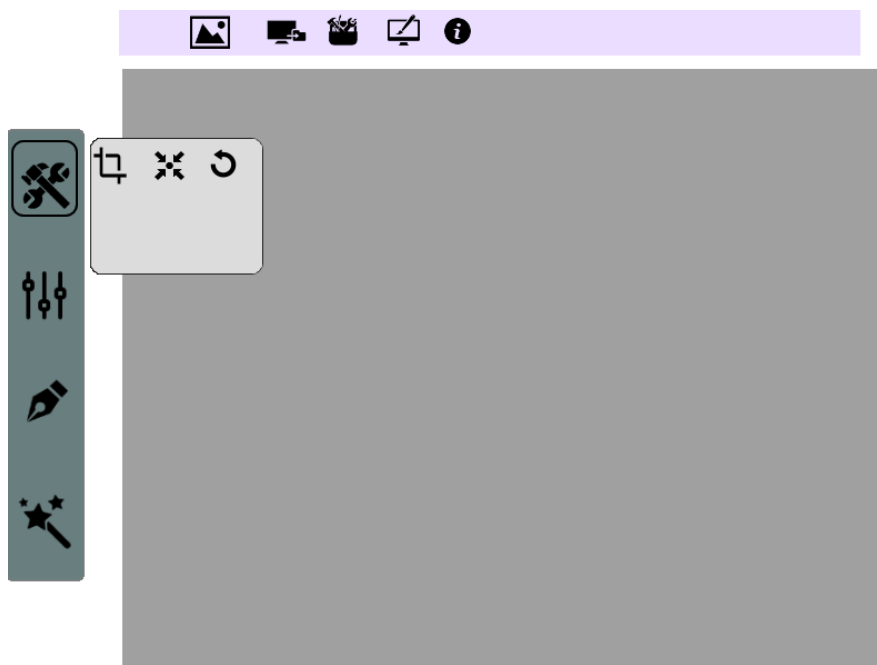


Figura 32 - MockUp do Layout (Sub-Menu)

3.1.5 Website oficial do projeto

Como auxiliar e informativo tanto para quem está a realizar o projeto como para o quem está a orientar, foi criado um sítio com todos os recursos relacionados com o projeto: objetivos do projeto, digramas, bibliografia relacionada, tecnologias e até uma seção que descreve o desenvolvimento que foi feito semanalmente.

Pode aceder através do seguinte endereço: <http://www.dei.isep.ipp.pt/~i120989/thesis/>.

De acordo com a seguinte figura, o website encontra-se dividido em 6 secções, cada qual representada por uma cor e ícone:

“Acerca de” (verde), “RoadMap” (Preto), “Software” (Castanho), “Livros” (Laranja), “Libraries & Frameworks” (Amarelo) e “Projeto” (Azul).

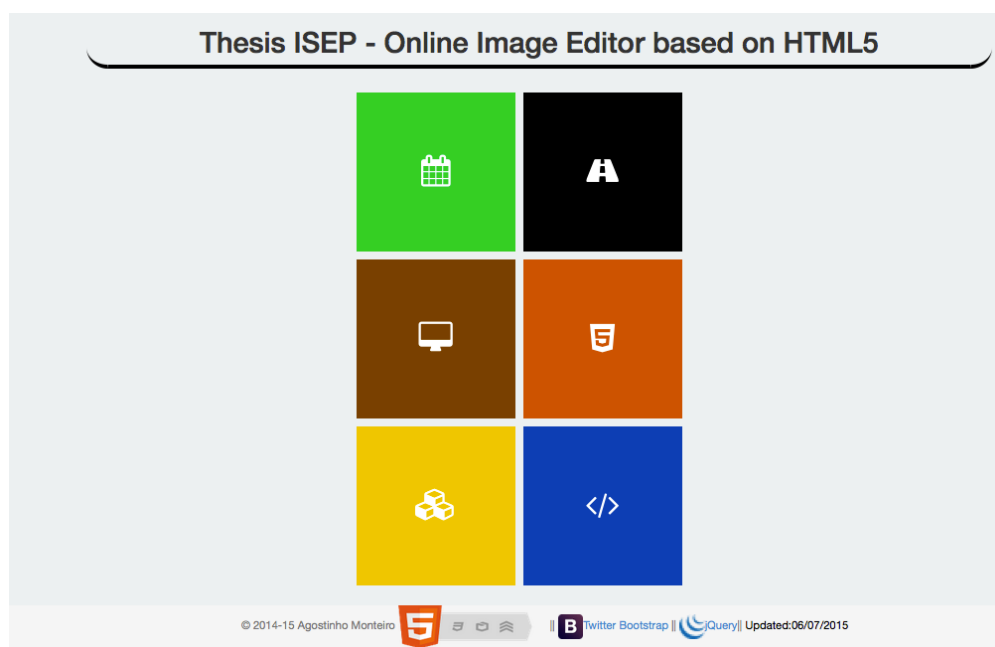


Figura 33 - Website Oficial do Projecto

Cada área representa alguns dos capítulos apresentados neste documento e cada seção representa:

- “Acerca de”: são apresentados os objetivos da tese, as fases em que a compõe bem como o respetivo mapa de Gantt;
- “RoadMap”: é apresentado o que foi feito semanalmente desde o início do estudo até à escrita do documento final da tese;
- “Software”: são apresentados alguns programas existente no âmbito da tese, existente no mercado:
- “Livros”: é apresentada alguma bibliografia fundamental para o desenvolvimento da aplicação, websites de referência bem como exemplos práticos da tecnologia a ser utilizada no protótipo;
- “Libraries & Frameworks”: são apresentadas quais as bibliotecas ou frameworks que existem para acelerar e melhor o desenvolvimento do protótipo bem como outras relacionadas as tecnologias apresentadas neste documento, como SVG e Adobe Flash.
- “Projecto”: são apresentado as fases, mais específicas do projeto, através do mapa de gantt, bem como links que associados a cada fase de desenvolvimento do protótipo, desde lançamento alfa do protótipo à versão Gama, versão de apresentação / defesa da Tese.

3.2 Protótipo

No segundo subcapítulo, é apresentado ilustrado o protótipo que foi desenvolvido após análise anteriormente referida, desde organização do Menu, às tecnologias utilizadas.

É igualmente feita a descrição geral dos módulos em que se desenvolve o protótipo.

Por último é apresentada a sua avaliação, feita por um grupo de pessoas

3.2.1 Estrutura de Navegação

Nesta secção, conforme se pode visualizar na figura abaixo será descrita a estrutura de navegação do sistema bem como a forma como as funcionalidades estão agrupadas.

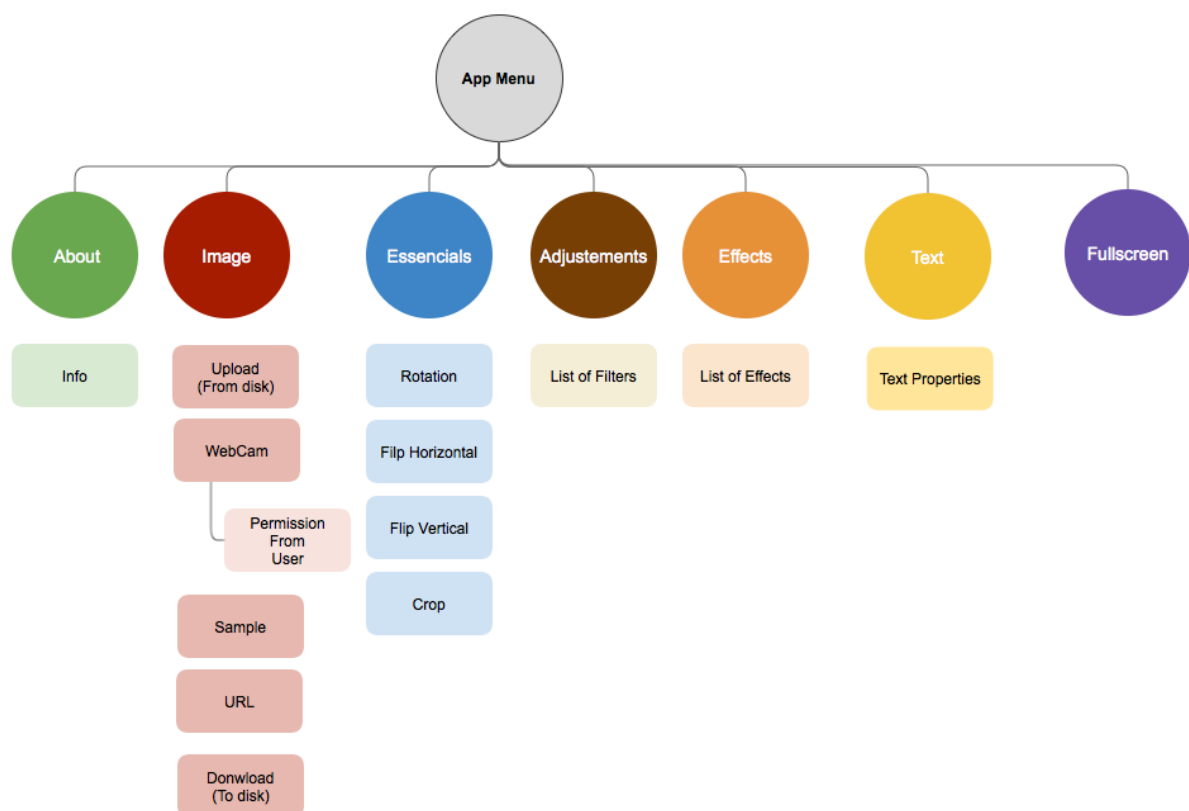


Figura 34 - Estrutura de Navegação da Protótipo

- Menu “About”: contém informação acerca da aplicação como versão bem como tecnologias / frameworks envolvidas.
- Menu “Image”: após seleccionar imagem, aparecerão 5 opções ao utilizador de carregar / inserir uma imagem
- Menu “Essenciais”: contém ferramentas para execução de operações rápidas sobre a imagem seleccionada
- Menu “Adjustments”: é disponibilizada uma listagem de filtros de forma a permitir a melhorar ou descaracterizar a imagem, de forma livre
- Menu “Effects”: é disponibilizada uma listagem de efeitos, constituído dois ou mais filtros com valores predefinidos.
- Menu “Text”: possibilita a inserção de texto bem como a alteração de propriedades como stroke, tamanho e cor.
- Menu “Fullscreen”: funcionalidade que permite colocar aplicação de modo a ocupar o ecrã por inteiro.

3.2.2 Tecnologias e Software

No capítulo Estado da Arte foram apresentadas algumas tecnologias relevantes, que poderiam ser encaradas como parte do projeto em causa ou que são concorrentes entre si.

As tecnologias seleccionadas estiveram como base além do estudo feito sobre tecnologias apresentadas, esteve também com base existentes ou não existentes nas aplicações apresentadas no estado da arte.

Para o desenvolvimento do protótipo foi utilizado as seguintes tecnologias e Frameworks ou bibliotecas:

- HTML5: para além ser utilizado os novos elementos (tags), é utilizada mais especificamente a seguintes APIs:
 - File Reader API, para carregamento da imagem do disco;
 - getUserMedia/Stream API, para aceder à câmara do Utilizador após respetiva autorização;
 - Canvas e respetiva API, elemento onde é feita a edição e manipulação da imagem através de HTML5 Canvas API do elemento em causa.
- CSS3: através desta tecnologia são feitas a formatações da página HTML bem como a animação para os menus de navegação.

- JavaScript: Linguagem responsável pela comunicação entre os elementos HTML, CSS e sobretudo o mais relevante, como via de comunicação entre ações do utilizador e elemento Canvas.

- Bibliotecas ou Frameworks, descritas no capítulo Tecnologias:
 - jQuery: Acesso mais eficiente ao DOM;

 - CamanJS: utilizada para aplicação de filtros e efeitos;

 - HammerJS: utilizada para eventos touch no elemento canvas, mais propriamente nos dispositivos móveis, a funcionalidade de zoom in ou Zoom out.

 - Twitter Bootstrap (v3.2): Utilizada para o desenvolvimento das interfaces gráficas e respetivos componentes, em conjunto com jQuery .

- Principal programa para o desenvolvimento:
 - Netbeans: é um IDE (Integrated Development Environment) open-source e gratuito. Atualmente pertence à Oracle e destina-se ao desenvolvimento de aplicações para desktop, smartphone e Web. Neste programa é possível desenvolver em linguagens de programação desde Java, C/C++. Relativamente ao Desenvolvimento Web, permite o desenvolvimento de aplicações HTML5 (HTML, JavaScript e CSS) bem como em PHP. Tem ainda a vantagem de suportar Frameworks como jQuery, Twitter Bootstrap. Atualmente está na versão 8 e disponível em qualquer sistema operativo desktop.

3.2.3 Plataformas

Este protótipo como sendo Web-Based pode ser utilizado em qualquer sistema operativo que tenha pelo menos instalado um web Browser, moderno e atualizado na sua versão mais recente, incluído também os sistemas móveis, visto que os elementos constituintes da interface gráfica possibilitarem a sua adaptação a distintos ecrãs.

Mesmo utilizando a versão mais recente, não é completamente garantido que o utilizador usufruirá completamente das funcionalidades ou que não ocorrerão erros, pois existe uma grande vasta de Browsers, sendo os principais Mozilla Firefox, Google Chrome, Opera, Safari e IE (Internet Explorer). Cujo dois últimos browsers mencionados não permitirem ao utilizador usufruir de certas funcionalidades, que serão indicadas mais à frente.

3.2.4 Cronograma

Relativamente à calendarização do desenvolvimento do projeto, está dividida em várias etapas. Este mapa foi sendo atualizado à medida ou caso se justificasse e conforme podem visualizar tanto na imagem abaixo como no Website oficial do projeto, a versão final do cronograma.

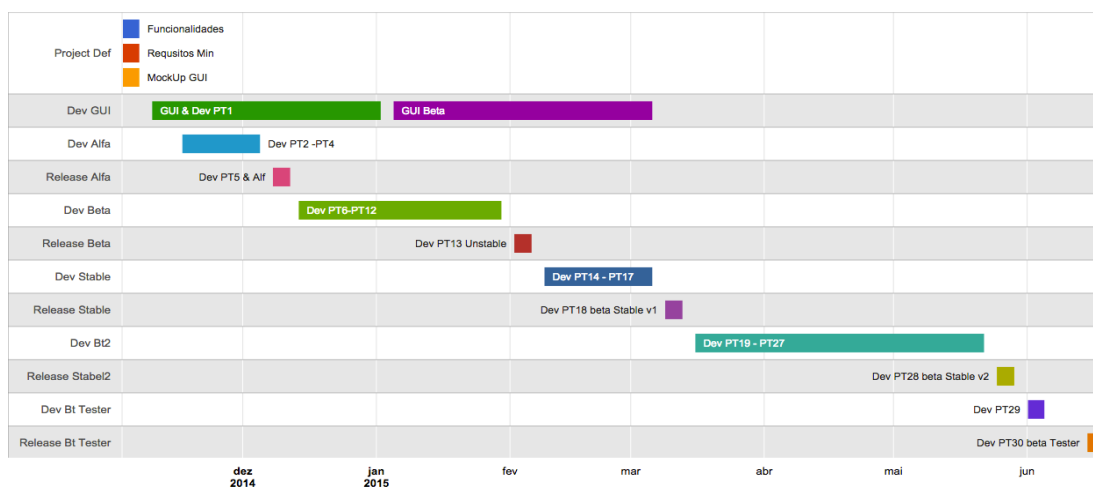


Figura 35 - Cronograma de Desenvolvimento

Após a fase de estudo de mercado à volta das aplicações existentes relativamente ao tema do projeto, bem como à pesquisa de tecnologias existentes para o desenvolvimento do mesmo, seguiu-se numa primeira fase a definição técnica do projeto como funcionalidades, qualificar as funcionalidades listadas, o respetivo desenho tanto visual como estrutural do protótipo.

Na segunda fase, Dev Gui, antes do desenvolvimento interface gráfica, foi escolhida a respetiva framework, entre Twitter bootstrap ou Zurb Foundation, bem como da IDE. Após estas escolhas, é desenvolvido a interface gráfica com a respetiva framework escolhida, Twitter Bootstrap. Esta fase prolonga-se à medida que se adicionam e desenvolvem novas funcionalidades.

A terceira fase, Fase Dev, está dividida em várias sub-fases desde alfa até testers, cada uma com a indicação da semana de lançamento (release) a ser demonstrada ao orientador, com a exceção da última fase que é aberta a pessoas que testam o protótipo, numa fase mais avançada. Nesta fase já está escolhida a tecnologia com que será programada o protótipo com as respetivas frameworks e APIs, anteriormente mencionadas.

3.2.5 Módulos

Nesta secção é apresentada, de forma geral, a lógica do desenvolvimento deste protótipo em termos de módulos e quais as funcionalidades estão agregadas a esses módulos.

3.2.5.1 Módulo GUI

Este módulo, GUI (*Graphic User Interface*), contém o desenvolvimento da interface gráfica. Nomeadamente tudo que diga respeito às ações do utilizador, desde ações com o rato no caso estar a utilizar um portátil ou desktop, ao evento que evolue o toque com as mãos, no caso de *Tablet* ou *Smartphone*. São também definidos as formas do menu, no caso de dispositivos móveis e naqueles dispositivos de maior dimensão e o tamanho do elemento Canvas. Dado que este elemento não possibilita a adaptação das suas dimensões ao dispositivo de forma autónoma, foi necessário fazê-la através de JavaScript.

3.2.5.2 Módulo Ficheiros

Este módulo relaciona-se com o carregamento da imagem selecionada por parte do utilizador. E consequentemente colocado no elemento responsável por visualização da imagem bem como do exportar / *download* da imagem.

O módulo é composto pelas seguintes funcionalidades referidas no subcapítulo “Análise do sistema”, sendo elas:

- Upload de Imagem pelo Disco (obrigatória);
- Upload de image Sample do servidor (obrigatória);
- Upload de imagem por URL (opcional);
- Guardar imagem para o disco do utilizador (obrigatória);
- Captura de imagem através da Webcam (opcional);
- Criação de Url da imagem para o utilizador (opcional).

A implementação deste módulo, com as funcionalidades anteriormente referidas, permite várias formas de selecionar imagem, abrangendo a seleção no seu disco rígido, por um URL, por imagem já existente no servidor, até à captação de uma imagem através da Webcam.

Para guardar a imagem modificada na aplicação há a possibilidade de fazer o Download / guardar para o disco rígido ou criar um URL, que poderá incorporar num Website.

3.2.5.3 Módulo Essenciais

Neste módulo inclui funcionalidades, que com base no capítulo estado da arte, são categorizadas como básicas e/ou essenciais em aplicações deste tipo, no quotidiano de um utilizador. Elas direcionam-se a operações de rápida edição.

As que foram implementadas e indicadas anteriormente são:

- Rotação (obrigatória);
- Flip vertical e Horizontal (obrigatória);
- Crop (obrigatória);
- Zoom (in/out) (opcional);
- Pan (opcional).

É um dos módulos mais complexos que requereu mais tempo de desenvolvimento., devido às dificuldades e decisões foram precisas identificar e selecionar.

As três primeiras funcionalidades anteriores listadas inserem-se no grupo das funcionalidades obrigatórias. As funcionalidades mais complexas foram o *crop*, da categoria das obrigatórias, e *zoom*, da categoria das opcionais. As dificuldades associadas às anteriores abordar-se-ão no subcapítulo Implementação. A fim de executar as funcionalidades aqui mencionadas, com exceção da última (*Pan*), basta o utilizador pressionar um elemento HTML, input do tipo range, semelhante a um *slider*. Sendo este último, um dos novos elementos inseridos no HTML5.

As funcionalidades Rotação e *Flip*, através API Canvas e respetivos métodos existentes, contribuíram para que o seu desenvolvimento fosse mais fluido na associação a componentes / elementos HTML, incorporando-se estas também na categoria da funcionalidades obrigatórias.

3.2.5.4 Módulo Ajustamentos

Este módulo é relativo ao tratamento de imagem, mais concretamente à manipulação dos níveis de cor e contraste da imagem. O utilizador tem a possibilidade de realizar estas operações de aperfeiçoamentos recorrendo a um conjunto de filtros disponibilizado, fornecido e implementado pela biblioteca CamanJS. Com o controlo slider pode ajustar-se a quantidade adequada à imagem selecionada.

3.2.5.5 Módulo Efeitos

Este módulo possibilita ao utilizador também modificar imagem, de forma mais ou menos drástica, consoante a seleção dos efeitos existentes. Estes são predefinidos pela biblioteca CamanJS, que cada efeito é constituído por dois ou mais filtros com valores predefinidos. Contrariamente ao que acontecia no módulo anterior, em que era preciso selecionar e indicar uma quantidade a aplicar, neste módulo o utilizador apenas seleciona, visualiza e aplica, caso seja do seu agrado.

3.2.5.6 Módulo Texto

Este módulo permite não só inserir apenas texto, mas também selecionar e modificar propriedades como tamanho, fonte e aplicar fontes predefinidas ao texto antes dele ser inserido.

Após o utilizador definir o texto e clicar no elemento Canvas, aparecerá algum muito semelhante ao cursor do rato, que enquanto não for inserido um carácter.

É de realçar, que este foi um dos módulos com alguma complexidade, isto porque concordando, em parte, com as boas práticas da W3C [W3 (b), 2015], deve evitar-se implementar controlos de edição de texto através do elemento Canvas, por causa de não terem sido implementadas na API Canvas funcionalidades para esse efeito, como por exemplo o cursor do rato, os movimentos do teclado, a edição de texto em multilinha, etc.

Isto traduz-se numa grande desvantagem para o programador, que ocuparia a maior parte do seu desenvolvimento só para implementar de raiz, as várias capacidades / funcionalidades na edição de texto.

O código deste módulo é baseado num dos livros (Core HTML5 Canvas), sendo alterado conforme as necessidades da implementação removendo umas e adaptando outras.

3.2.5.7 Módulo Ecrã inteiro

Neste módulo encontra-se uma das funcionalidades apontada como opcional, que possibilita ao utilizador, através do respetivo ícone no menu ou pressionando a tecla Enter para preencher o ecrã por inteiro. E novamente através do ícone ou tecla Esc para sair do ecrã inteiro, mostrando novamente os menus e/ou barras que vêm com o Browser

É uma das novas API do HTML5, no desenvolvimento de aplicações, que somente os Browser mais recentes são capazes de usufruir.

3.2.6 Interface com Utilizador

Neste módulo é apresentado a interface gráfica do protótipo, que se adapta conforme as dimensões do dispositivo. Na descrição de cada módulo são apresentados exemplos específicos de cada funcionalidade.

Na primeira figura abaixo, encontra-se a forma como é apresentada o menu e a imagem no dispositivo de média e grandes dimensões, por exemplo em desktop. Em que no menu principal, aparece os ícones que representam menus desde informação sobre protótipo à menu de efeitos. Encontra-se também disponíveis as seguintes funcionalidades: Fullscreen e Zoom.

Para a uma transição mais fluida entre o menu principal e submenus, é executada animação de fade-in, através do módulo de animação do CSS3.

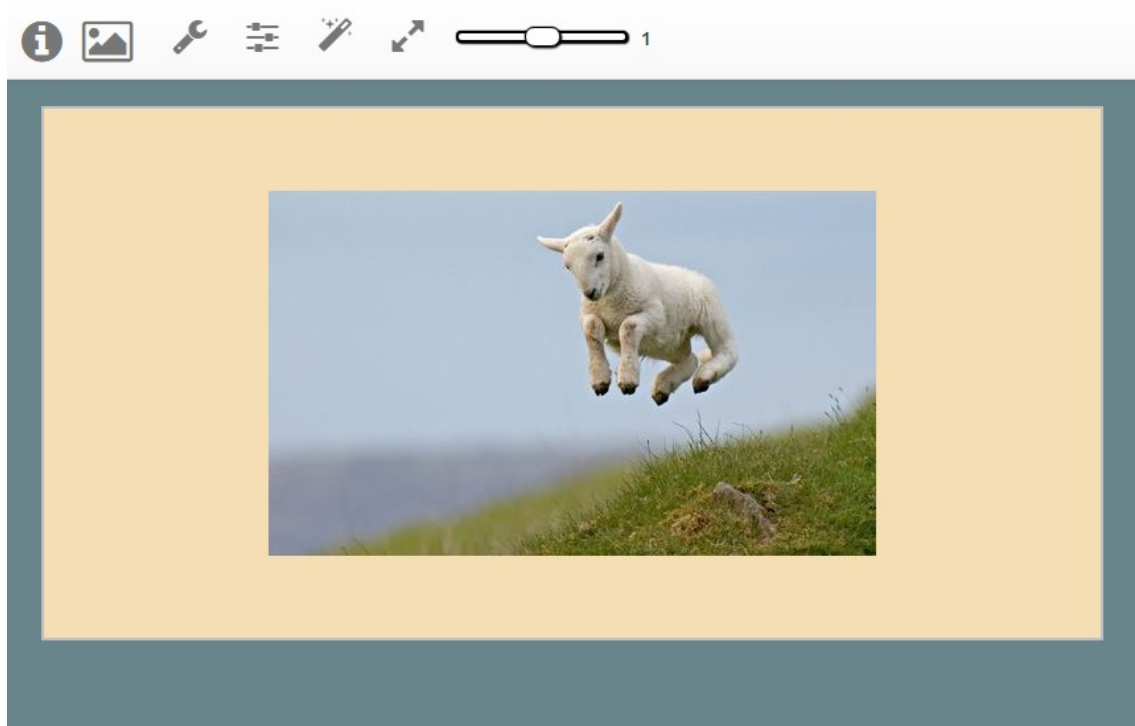


Figura 36 - User Interface: Desktop

Na próxima figura abaixo, demonstra a disposição no caso dispositivos móveis. Neste caso menu o principal somente mostra os ícones que representam, informação sobre protótipo, carregamento da imagem e ícone para possibilita mostrar o menu secundário.



Figura 37 - User Interface: Dispositivos Móveis

Na figura abaixo, ilustra como aparece o menu secundária com o respetivos ícones. Contrariamente ao desktop que basta passar o rato para identificar o ícone, neste caso aparece a identificação textual dos respetivos ícones.

Esta funcionalidade de menu secundária é possível através da framework Twitter bootstrap, que utiliza o CSS3.

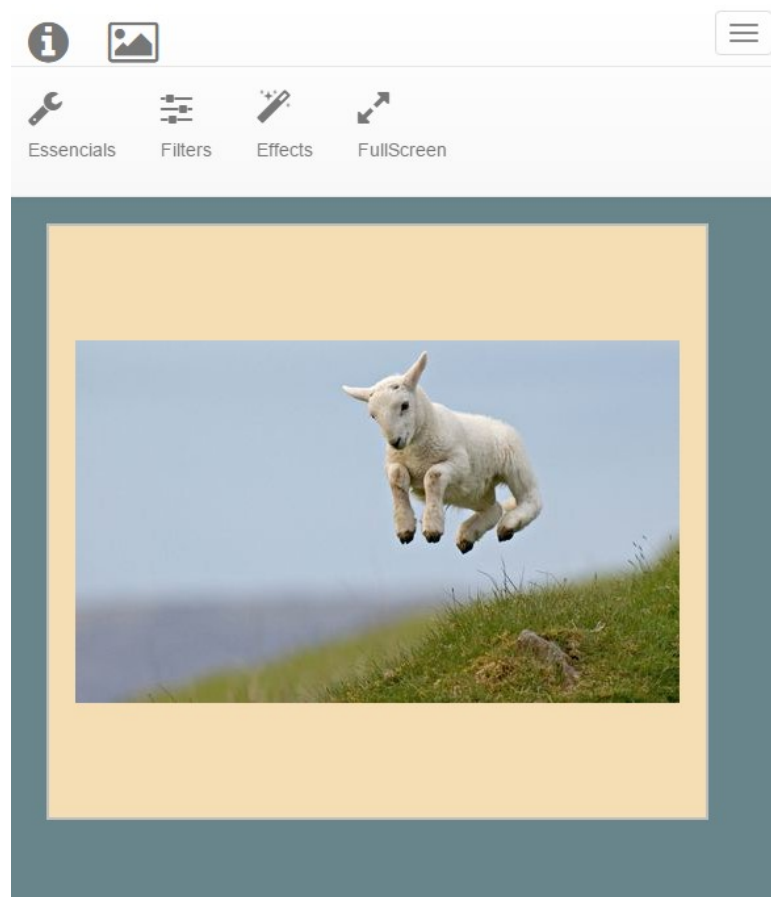


Figura 38 - User Interface: Dispositivos Móveis (Menu)

3.2.7 Avaliação

Neste capítulo são apresentados os dados resultantes dos testes efetuados por um conjunto de pessoas, ao protótipo. Para a realização deste testes e o preenchimento dos inquéritos para a avaliação do protótipo, foi estabelecido o prazo de uma semana, seguindo o endereço: <http://www.dei.isep.ipp.pt/~i120989/thesis/testers.html> .

3.2.7.1 Website dedicado Avaliação

Para os inquiridos testarem o protótipo e preencherem os respetivos inquéritos, foi desenvolvido um Website com o propósito de centralizar tudo que diga respeito à fase de teste do protótipo, que inclui instruções ou recomendações do preenchimento do inquérito bem como referentes ao protótipo.



Figura 39 - Website direcionado a testers

Conforme se pode visualizar na imagem acima, o Website está dividido em 4 tabs: Instruções, formulário Perfil, vídeo guião & Aplicação e Formulário de Aplicação

Na primeira tab, Instruções, são apresentadas informações relevantes, referentes à aplicação bem como referentes à maneira de proceder a avaliação, que é seguindo a ordem como está identificadas nos tabs. Finalmente está indicado a data limite do preenchimento dos inquéritos e de efetuar o teste ao protótipo.

Na segunda tab é apresentado um inquérito ao utilizador, com o propósito conhecer melhor o utilizador em ferramentas / Software de edição e manipulação de imagens tanto nos desktop como em smartphones.

Na terceira tab, vídeo guião & Aplicação, é apresentado ao utilizador a possibilidade de antes de testar protótipo, visualizar um vídeo demonstrativo, de 3 minutos, da aplicação, percorrendo todos os menus aí existentes. Caso visualize o vídeo ou não, é disponibilizado um botão que abrirá numa nova tab no Browser do utilizador, que executará protótipo a testar.

Por último, a tab Formulário Aplicação, como próprio nome indica, é apresentado um inquérito que diz respeito à utilização da aplicação e a respetiva avaliação feita pelo utilizador.

3.2.7.2 Ferramentas de Recolha de Dados

Como foi referido anteriormente, o teste do protótipo foi efetuado numa duração de uma semana, mais especificamente entre os dias 15/06/2015 a 19/06/2015, através de um conjunto de pessoas, dentro do ISEP. Mais precisamente, alunos que estão inscritos dentro dos três ramos do mestrado de Engenharia, foi enviado um email com o pedido de colaboração, indicado o endereço do website anteriormente referido. Em anexo 1 e 2 encontram-se os respetivos inquéritos.

Para auxiliar ao estudo de perfil de utilizador bem como á avaliação do protótipo foram utilizadas duas ferramentas, ambas pela mesma empresa Google, que são as seguintes: Google Analytics e Google forms.

Através de Google Analytics, é possível recolher informação, como por exemplo sistema operativo, país, dimensões de ecrã, a que acede a um determinado Website / aplicação, através um pedaço de código que funciona como tracker, que tem a responsabilidade de monitorizar o acesso e utilização do Website / aplicação. Com esta ferramenta foram criados dois trackers, em JavaScript, com a responsabilidade de acompanhar o acesso tanto ao Website anteriormente referido bem como o acesso à aplicação.

Na ferramenta Google Forms, foram criados os dois formulários que foram incorporados no Website anteriormente apresentado, onde um deles diz respeito ao perfil de utilizador no que toca a feramente de edição e manipulação de imagem, enquanto o segundo formulário diz à utilização e avaliação do protótipo. As respostas dos utilizadores são guardas numa ferramenta semelhante a uma folha de cálculo, onde se pode criar automaticamente, com base nas respostas, pode-se processar os dados, através de gráficos.

3.2.7.3 Análise dos dados do inquérito de perfil de utilizador

O objetivo do primeiro inquérito consiste em conhecer os inquiridos em termos de conhecimento em Software de edição e manipulação de imagem em diversas plataformas. Onde este primeiro inquérito está dividido em 4 seções fundamentais: geral, Desktop, Online e dispositivos móveis.

É de referir que número total de participantes neste primeiro inquérito é de 33 pessoas.

No contexto geral, que é direcionado a conhecer características da pessoa, conforme se pode verificar na imagem abaixo. Dado enquadramento onde foi lançado o inquérito, num estabelecimento de ensino da área tecnológica, vem a confirmar o esperado, que é os inquiridos são maioritariamente jovens e do sexo masculino, apontando a idade mais especificamente entre 21 e 25 anos.

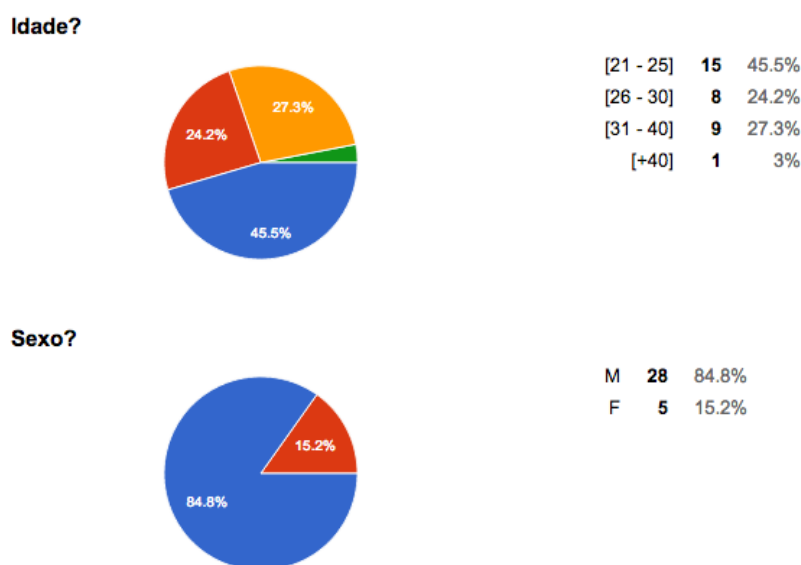


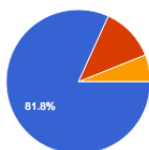
Figura 40 - Inquérito 1: Geral

Na segunda parte, Desktop, são feitas questões de sistema operativo que mais trabalha no seu dia-a-dia bem como se utilizador Software de edição de imagem. Relativamente a questão o sistema operativo os resultado encontram-se em sintonia com a generalidade das estatísticas acerca deste tópico, que o sistema operativo Windows, mesmo em instituições tecnológicas, é altamente escolhido e com grande vantagem relativamente a outros sistemas operativos.

Na questão relativa a Software de imagem, sendo que a questão foi feita a universo de pessoas a frequentar um Curso na área de Informática, o resultado não surpreende, com mais de 80 % das pessoas a apontar já manuseou este tipo de Software quer a nível académico e / ou quer a nível profissional.

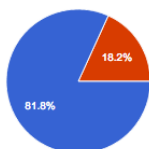
Ainda neste tipo de plataforma, mais concretamente Offline, é apontado uma lista onde é questionado quais dos Softwares em concreto já trabalhou, onde que o resultado, devido também à grande popularidade que Software da Adobe tem, a maioria pelo menos testou Adobe Photoshop, que é um Software pago.

Qual o Sistema Operativo Desktop que Utiliza?



Microsoft Windows	27	81.8%
Apple	4	12.1%
Outro Unix Based	2	6.1%

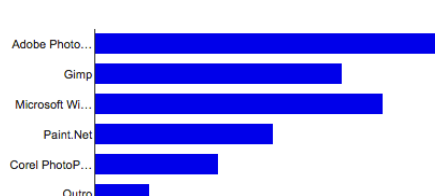
Já trabalhou, a nível académico ou Profissional, Editores de imagem?



Sim	27	81.8%
Não	6	18.2%

Software & SO - Desktop (Cont.)

Quais dos seguintes editores já trabalhou ou testou?



Adobe Photoshop	25	92.6%
Gimp	18	66.7%
Microsoft Windows - Paint	21	77.8%
Paint.Net	13	48.1%
Corel PhotoPaint	9	33.3%
Outro	4	14.8%

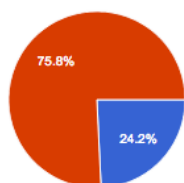
Figura 41 - Inquérito 1: Desktop

Em segundo lugar, surpreende ou não, dependente da perspetiva, como sendo Software gratuito e como vindo de raiz no universo do sistema operativo Windows, a Microsoft Paint que é direcionado para execução tarefas mais elementares a nível de edição de imagem, com 77,8%.

Em terceiro lugar, sendo este Software também gratuito, disponível em qualquer sistema operativo, sobre tudo como sendo Open-Source, podendo quase equiparar-se a Software pago anteriormente referido da Adobe, encontra-se o GIMP com cerca de 67%.

Na terceira fase, diz respeito a utilização deste tipo de Software nativo em dispositivos móveis, sendo que somente 24,2 % dos inquiridos já instalou e testou em Smartphone ou em Tablets. Daqueles que testou, foi apresentado uma curta lista de Software existente para este tipo de dispositivos, onde um software responsável pela AutoDesk e pela Adobe indicado na imagem abaixo, Autodesk Pixlr a ser o mais utilizado pelo utilizador. É de referir daqueles que escolheu a opção outro, colocou que utiliza o que vem de raiz pelo dispositivo.

Já testou algum editor de imagem num smartphone / tablet?



Sim	8	24.2%
Não	25	75.8%

Software & SO - Smartphone & Tablet (Cont.)

Quais dos seguintes editores tem ou já teve no seu smartphone ou Tablet?



Fotor	1	14.3%
Adobe Pixlr	5	71.4%
Outro	2	28.6%

Qual o Sistema Operativo do seu smartphone or Tablet?



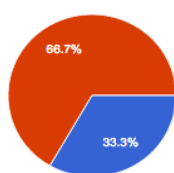
Android	7	87.5%
iOS	1	12.5%
Windows Phone / Mobile	0	0%
Outro	0	0%

Figura 42 - Inquérito 1: Sistemas Móveis

Relativamente ao sistema operativo neste tipo de dispositivos, o resultado não contraria globalmente é apresentado quer pelo Google quer por operadoras, que o sistema operativo do Google, o Android é o mais utilizado.

Na última fase do inquérito 1, que diz respeito a Software de edição na plataforma Online, com somente cerca de 33 % dos inquiridos a indicar já ter utilizado programas deste tipo online. Onde daqueles que responderam que sim, na pergunta associado a lista de Software existente, a maioria indica que entre um empate técnico, ter utilizado o Autodesk Pixlr e outro.

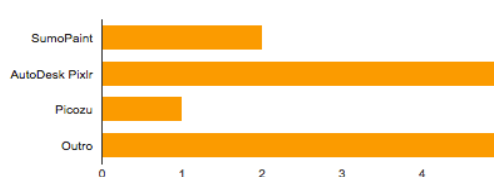
Já trabalhou ou testou algum editor de imagem Online?



Sim	11	33.3%
Não	22	66.7%

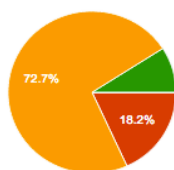
Software & SO - Online (Cont.)

Quais dos seguintes editores Online já trabalhou ou testou?



SumoPaint	2	18.2%
AutoDesk Pixlr	5	45.5%
Picozu	1	9.1%
Outro	5	45.5%

Qual o Browser que utiliza mais neste momento?



IE	0	0%
Mozilla Firefox	2	18.2%
Google Chrome	8	72.7%
Safari	1	9.1%
Opera	0	0%
Outro	0	0%

Figura 43 - Inquérito 1: Software Online

Relativamente à ferramenta ou Browser que os utilizadores mais utilizam, vem ao encontro da estatística existente na internet, com a maioria a preferir o Browser Google, o Chrome, com 72,7 %. Em segundo encontra-se o Browser Mozilla Firefox.

3.2.7.4 Conclusão ao inquérito de perfil de utilizador

No primeiro consiste conhecer melhor os inquiridos em termos de conhecimentos de programas de edição e manipulação tanto plataforma desktop como em dispositivos móveis. Neste primeiro inquérito retira-se 4 grandes ilações.

Primeiro, maioria dos utilizadores em desktop utilizada a sistema operativo Windows. E em sistemas móveis, o Android ainda tem grande impacto, com maioria a preferir este sistema operativo.

Segundo, apesar de existir Software open-source e gratuito, como GIMP que ficou em segundo na escolha. A maioria prefere ou já testou o Adobe Photoshop para edição de imagens.

Em terceiro, a maior a preferir, ou nunca ter utilizado o seu dispositivo móvel para edição ou manipulação imagens. Onde aqui, no que toca a utilização de um determinado Software ou aplicação, é mais equilibrado devido no sistema móvel android haver uma boa variedade de aplicações gratuitas.

Em quarto e em sintonia com a última afirmação, mas no que toca a Software ou aplicação via Web / Browser, a maioria nunca ou quase nunca utilizou para alterar imagens, Software ou aplicação via Browser, o Browser dominante é o Google Chrome.

Para o projeto, contribuiu informações como o Browser mais utilizado, tipo ou dimensões dos dispositivos e quais os programas mais utilizados.

No caso do Browser mais utilizado é importante porque, de acordo com Website HTML5 Test [HTML5test, 2015], o Browser Google Chrome, juntamente com Opera e Mozilla Firefox, tem maior suporte à tecnologia HTML5, o facilita tanto para quem está a desenvolver como para quem está a utilizar.

Igualmente importante, era interessante saber qual o tipo ou dimensões de dispositivos mais usados podendo dar maior prioridade a estes na altura do desenvolvimento, sendo primeiramente para dispositivos de média e grande dimensão.

Por último, e não menos importante, a informação sobre os quais os programas de edição mais utilizados, permite desenvolver uma interface gráfica acessível e simples a estes utilizadores.

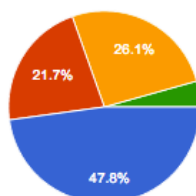
3.2.7.5 Análise dos dados do inquérito de Avaliação do protótipo

O segundo inquérito diz respeito à avaliação feita por parte de quem testou aplicação, é dividida em 3 grandes seções: Geral, desktop e Mobile. Antes de testarem a aplicação os inquiridos tiveram acesso a um vídeo demonstrativo da aplicação. Número de participantes neste segundo inquérito foi 23 pessoas, o que contrasta com o primeiro inquérito houve um decréscimo de cerca de 30 % ou seja, menos 10 pessoas.

Na primeira fase, mais precisamente Geral, com o mesmo objetivo do primeiro inquérito, o resultado dos intervalos em causa não se altera, estando a idade da maioria compreendida entre 21 e 25 anos, sendo a maioria dos inquiridos do sexo masculino.

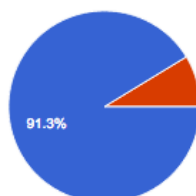
A acrescentar as questões anteriores, é colocado a questões sobre a visualização do vídeo guião. Onde respondeu positivamente que foi visualizado o vídeo.

Idade?



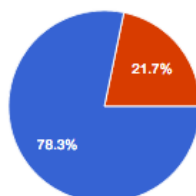
[21 - 25]	11	47.8%
[26 -30]	5	21.7%
[31-40]	6	26.1%
[+40]	1	4.3%

Sexo?



M	21	91.3%
F	2	8.7%

Visualizou o video de demonstração da aplicação?



Sim	18	78.3%
Não	5	21.7%

Figura 44 - Inquérito 2: Geral

Na segunda parte do segundo inquérito, à avaliação da aplicação na plataforma desktop, onde conforme se pode visualizar na imagem seguinte, o resultado referente ao sistema operativo onde foi testado aplicação, é uma continuidade resultado do inquérito anterior, que vem a confirmar o sistema operativo da Microsoft como o preterido da maioria dos inquiridos, com 87%.

Relativamente ao Browser onde foi testado aplicação, em sintonia com o primeiro inquérito no que toca ao Browser mais utilizado, o Google Chrome é o Browser de eleição dos inquirida para testar aplicação, com 73,9 %, onde conforme se pode visualizar nos dados seguintes, pode-se retirar a ilação que houve utilizadores que testaram pelo menos em mais que um Browser, conforme era o caso desejado.

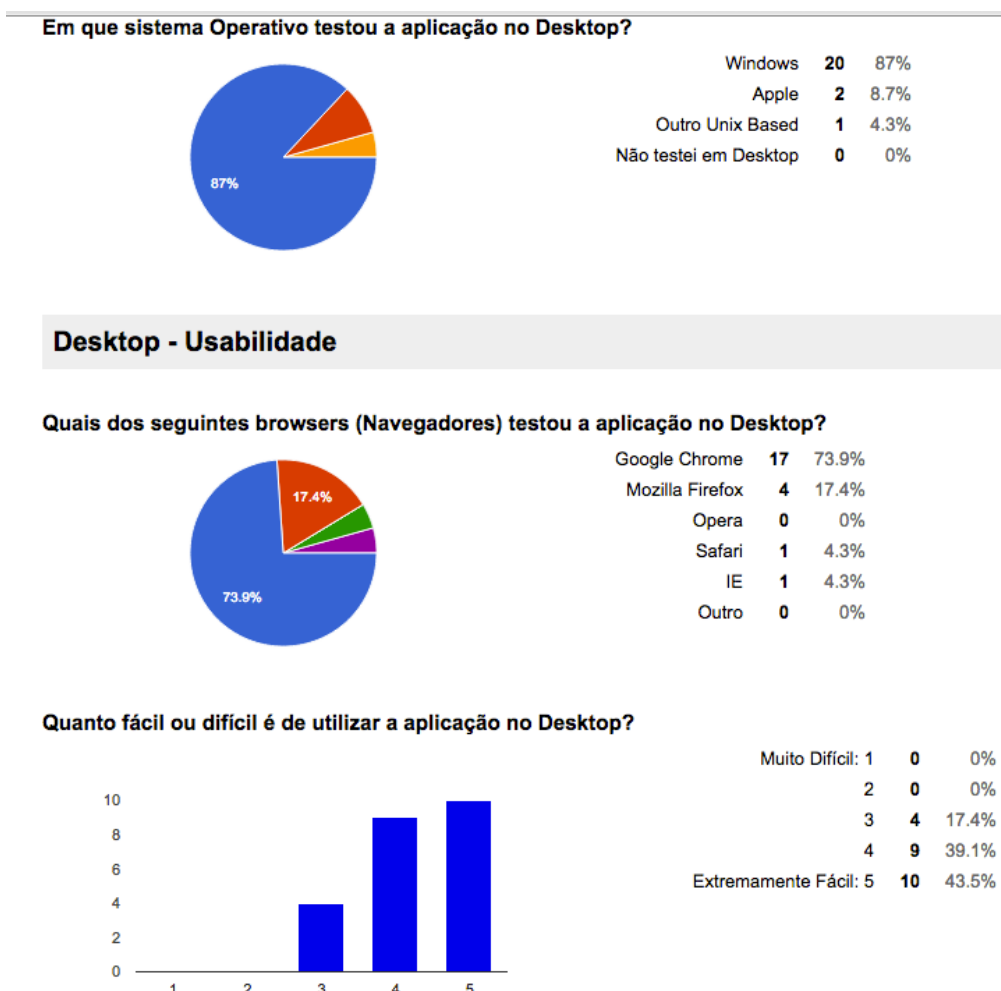


Figura 45 - Inquérito 2 – Teste em Desktop

Relativamente a questão de usabilidade da aplicação na plataforma desktop, a maioria aponta que aplicação como Extremamente fácil, ou seja, os inquiridos é de supor que a interface gráfica e acções associadas iam ao encontro da maioria dos inquiridos e fácil compreensão.

Ainda em termos de usabilidade, mais concretamente nível visual, quase a maioria dos inquiridos, ou seja, 43,5% acham bem atraente e aplicativa à medida que se utiliza. Conforme se pode ver na imagem abaixo.

Quanto é visualmente atraente é a interface Gráfica no Desktop?

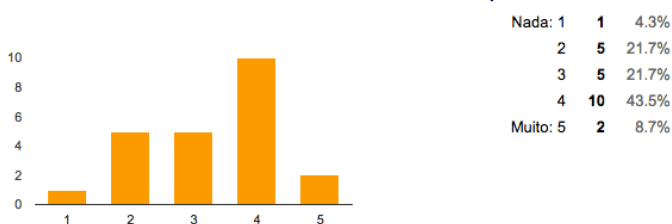
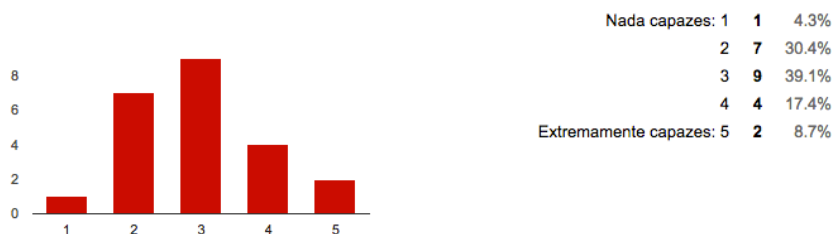


Figura 46 - Inquérito 2: Teste Visual em Desktop

Quando questionados se as funcionalidade existentes seriam suficientes para editar imagens, maioria, com 39,1% aponta que sim contudo 30 % indica que poderia haver mais.

Quanto a aplicação tem funcionalidades capazes de colmatar as suas necessidades?



Conseguiu executar todas as tarefas desejadas?

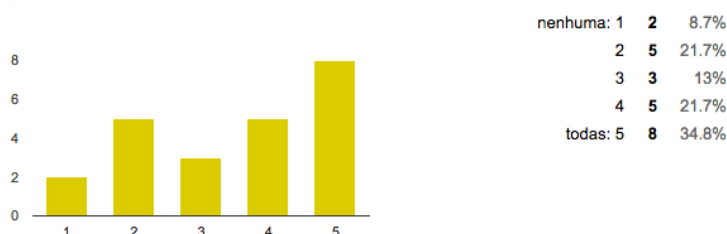


Figura 47 - Inquérito 2: Funcionalidades Implementadas em Desktop

Quanto á execução das funcionalidades existentes, grande parte dos inquiridos conseguiu aplicar todas as que foram pedidas. Alguns, com 21,7%, indicam que quase todas ou nem todas, dando a entender como era de esperar, ainda haver falhas em algumas funcionalidades.

Relativamente a funcionalidades gostariam que fossem implementadas, em primeiro fica o Undo/ refazer com grande importância aos inquiridos. Em segundo encontram-se duas funcionalidades, *Drag & Drop* e Combinar múltiplas imagens, com 60%. Em terceiro lugar com 47,8% ficaram as funcionalidades de Texto e *Crop*. Onde estas mesmas também já na altura da versão testes já tinham sido implementadas desde primeira versão beta mas que devido a estarem incompletas ou algo instáveis para serem disponibilizadas, foi decidido esconder estas funcionalidades que serão demonstradas na versão gama.

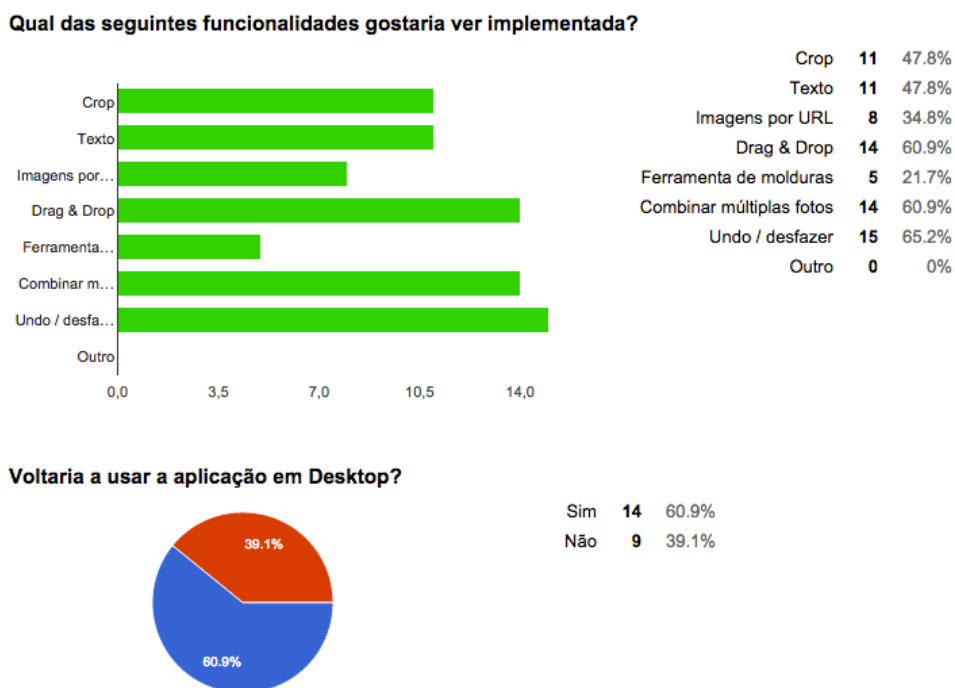


Figura 48 - Inquérito 2: Funcionalidades a Implementar em Desktop

Na segunda parte da imagem acima, a maioria dos inquiridos indica que futuramente a poderiam utilizar.

Na terceira parte do inquérito 2, é questionado aos utilizadores sobre a sua utilização em sistema móveis e respetivos Browsers.

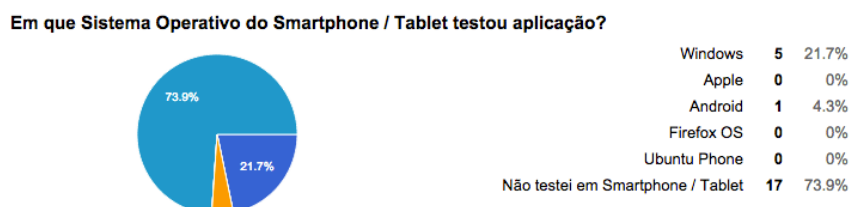


Figura 49 - Inquérito 2: Teste em dispositivos móveis

Naqueles utilizadores que testar nos seus respetivos dispositivos móveis, foi questionado à semelhança em desktop, desde usabilidade ate funcionalidades que gostariam ver neste tipo de dispositivos. Na figura acima existe uma anomalia face ao inquérito 1, no mesmo contexto. Em que no tracker do Google analytics (Figura abaixo) ilustra quais os sistemas operativos móveis que acederam ao protótipo, o que não inclui Windows Phone/Mobile.

Sistema operativo	Sessões	% Sessões
1. Android	2	66,67%
2. iOS	1	33,33%

[ver relatório completo](#)

Figura 50 - Inquérito 2: Tracker Google Analytics

Conforme se pode visualizar na imagem apresentada abaixo, á semelhança dos resultados do inquérito anterior, o Browser Google Chrome em grande vantagem relativamente aos outros Browsers com cerca de 67% dos votos.

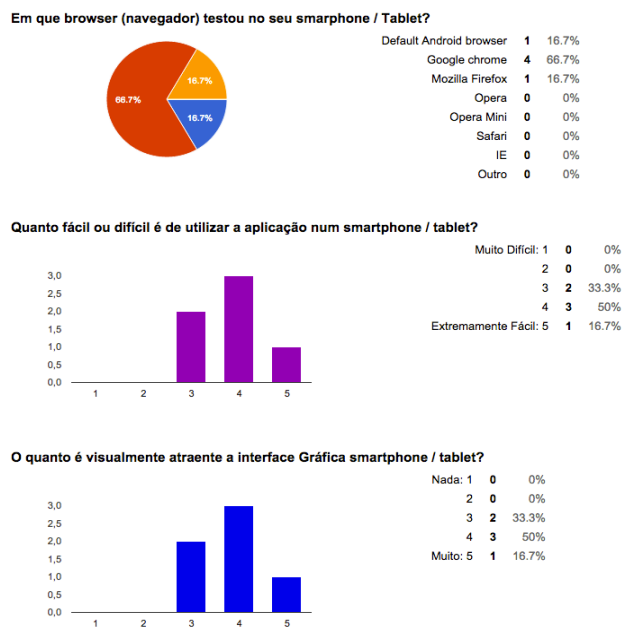


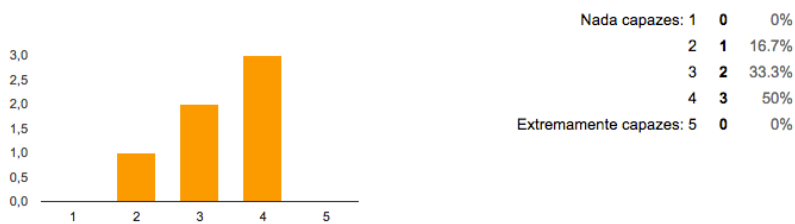
Figura 51 - Inquérito 2: Teste visual em Dispositivos Móveis

Em termos de utilização, a maioria dos votos incide entre o razoável e bom, que tende a dizer que não estiver com grandes dificuldades em encontrar as funcionalidades a aplicar, estando bem organizada a aplicação.

Na última parte da imagem acima, os resultados são semelhantes ao que pode indicar que gostaram mas poderia ser mais apelativo.

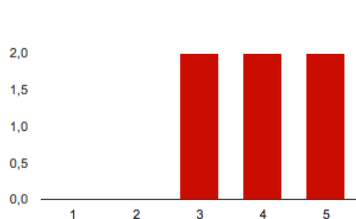
Na seção do inquérito 2, que diz respeito às funcionalidades e sua aplicabilidade, conforme se pode visualizar na imagem abaixo, os respectivos resultados.

O quanto a aplicação tem funcionalidades capazes de colmatar as suas necessidades?



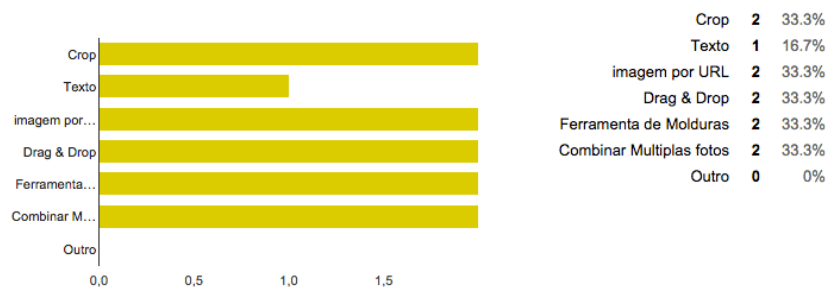
Nada capazes: 1	0	0%
2	1	16.7%
3	2	33.3%
4	3	50%
Extremamente capazes: 5	0	0%

Conseguir executar todas as tarefas desejadas?



nenhuma: 1	0	0%
2	0	0%
3	2	33.3%
4	2	33.3%
todas: 5	2	33.3%

Qual das seguintes funcionalidades gostaria ver implementada?



Crop	2	33.3%
Texto	1	16.7%
imagem por URL	2	33.3%
Drag & Drop	2	33.3%
Ferramenta de Molduras	2	33.3%
Combinar Múltiplas fotos	2	33.3%
Outro	0	0%

Figura 52 - Inquérito 2: Funcionalidades em dispositivos Móveis

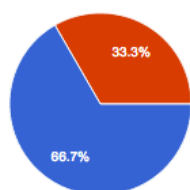
Somente metade dos inquiridos acha que são suficientemente boas para realizar o desejado naquele tipo de dispositivos. É de salientar que alguma acham que são insuficientes e que poderia haver mais.

Na segunda parte da imagem acima, é questionado se foram aplicadas todas as ações desejadas, os inquiridos, globalmente indicam que foram executadas de forma satisfatória todas as ações exercidas sobre a respetiva imagem.

No último tópico que se pode visualizar na imagem, deram relevância a maior das ferramentas que gostaria ver implementadas de igual forma, com exceção da ferramenta de texto.

Por ultimo, á semelhança, à parte relativa à desktop, maioria indica, sabendo que aplicação estar a fase de maturação, que futuramente voltaria a usar.

Voltaria a usar a aplicação num smartphone / tablet?



Sim 4 66.7%
Não 2 33.3%

Figura 53 - Inquérito 2: Uso futuro

3.2.7.6 Conclusão ao inquérito de avaliação do protótipo

No segundo inquérito, onde o foco principal é utilização do protótipo com o respetivo feedback, do qual se pode deduzir várias coisas.

À semelhança do inquérito, a maioria utilizou o sistema operativo Windows, através do Browser Google Chrome para testar o protótipo. Sendo que que a maioria achou de simples utilização tanto em desktop como é dispositivos móveis.

A maioria dos inquiridos conseguiu executar a tarefas pretendidas. Também no que toca a funcionalidades a implementar a maioria tanto em desktop como dispositivos móveis gostaria de ver a funcionalidade *Crop*.

Na altura da submissão do protótipo a avaliação, a funcionalidade *Crop*, estava instável, sendo decido não dar acesso a esta funcionalidade bem como a de Texto.

Neste inquérito, o importante a reter e que ajudou na evolução do protótipo foi o feedback relacionado com as funcionalidades mais relevantes, que para o inquirido são o *Crop* e Texto.

Que posteriormente, o principal foco do desenvolvimento foi aperfeiçoamento destas duas funcionalidades.

Também foi importante saber a nível de interface, o grau de dificuldade por parte do inquirido, que não ter havido grandes dificuldades de adaptação, não foram feitas grandes alterações.

3.3 Implementação

Nesta seção são narrados de forma mais técnica os módulos implementados, descritos de forma genérica, no subcapítulo Módulos. Dando mais relevância a certos aspetos, como dificuldades e decisões efetuadas ao longo do desenvolvimento do protótipo bem como apresentação de algum código implementado e figuras pertinentes para um melhor entendimento.

3.3.1 Módulo GUI

Este módulo é responsável pela interação do utilizador com a interface gráfica. Neste módulo é utilizado tecnologia CSS, mais concretamente a sua especificação mais recente: CSS3, através de utilização de animação responsável pela transição entre menus e funcionalidades.

Através da Framework Twitter Bootstrap é criado menu capaz de se adaptar a qualquer tipo de dispositivo, onde esta Framework utiliza funcionalidades mais recente da especificação CSS3, como *media queries*. Também através da Framework, é possível criar componentes como *Modal*, que é o aparecimento de algo semelhante a um *popup*, como poderá verificar na figura do módulo de texto, referente às propriedades do texto. Ou como aparecerá ao selecionar no ícone de imagem, com as respetivas opções de carregamento de uma imagem.

Na figura 50 encontra-se o estado inicial do menu, desde o menu *Image* até a funcionalidade de Zoom, com o respetivo *slider*.

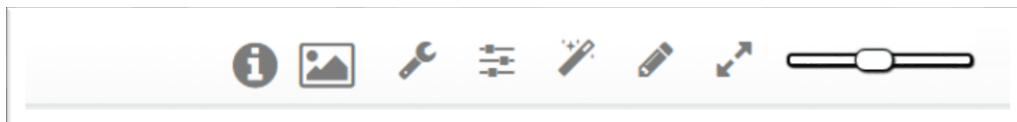


Figura 54 - Módulo Menu: Inicial

Através da Framework jQuery, é criada a dinâmica entre menus, que após selecionar o respectivo menu, por exemplo menu *Essenciais*, são escondidos os restantes menus iniciais e aparecerá os ícones das funcionalidades daquele menu, conforme se pode visualizar na figura 44. Para regressar ao menu inicial basta selecionar o botão de cor verde para aplicar a alteração ou o botão de cor vermelha para cancelar a alteração para posteriormente voltar ao menu principal.

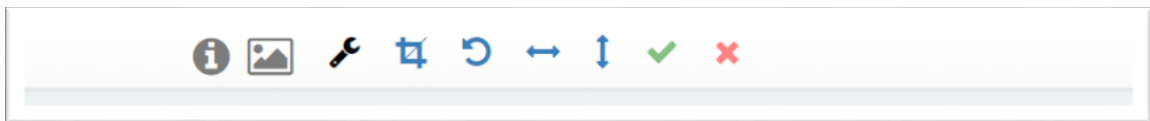


Figura 55 - Módulo Menu: Essenciais

Para o aparecimento das funcionalidades do menu Essenciais, pode-se visualizar a implementação da dinâmica através jQuery no lado direito da figura 45, onde são escondidos todos menus através de uma classe CSS fornecida pela Frameworks Twitter Bootstrap, e removida a Classe CSS que escondia o respectivo Submenu.

No lado esquerdo da figura 45 encontra-se desenvolvido através de CSS a animação que aplicada após selecionar o menu. A animação é feita em duas fases, onde a primeira fase, através de *keyframes*, é definido o estado inicial e o estado final, que neste caso, aparecerá inicialmente invisível.

```

custom.css
123 /* Dá indicação à keyframe o seu estado inicial e estado final */
124 @-webkit-keyframes fadeIn {
125   from {
126     opacity: 0;
127   }
128   to {
129     opacity: 1;
130   }
131 }
132 @-moz-keyframes fadeIn {
133   from {
134     opacity: 0;
135   }
136   to {
137     opacity: 1;
138   }
139 }
140 @keyframes fadeIn {
141   from {
142     opacity: 0;
143   }
144   to {
145     opacity: 1;
146   }
147 }
148 /* Animation fade-in */
149 .fade-in {
150   /* Coloca inicialmente o elemento invisível */
151   opacity: 0;
152   -webkit-animation-delay: 0.7s;
153   -moz-animation-delay: 0.7s;
154   animation-delay: 0.7s;
155 }
156
157 /* Chama a keyframe com o nome FadeIn, usando a propriedade da animação ease-in
158 somente uma vez */
159 -webkit-animation: fadeIn ease-in 1;
160 -moz-animation: fadeIn ease-in 1;
161 animation: fadeIn ease-in 1;
162
163 -webkit-animation-duration: 1s;
164 -moz-animation-duration: 1s;
165 animation-duration: 1s;
166
167 /* Abaixo mantém o estado final da animação após o sua colocação, ou seja com
168 opacidade 1 */
169 -webkit-animation-fill-mode: forwards;
170 -moz-animation-fill-mode: forwards;
171 animation-fill-mode: forwards;
172 }
173
174 ---
app.js
76
77 $( "#menuEssencial" ).click(function () {
78
79   $( "#li.menu" ).addClass("hidden");
80   $( "#li.essenciais" ).removeClass("hidden");
81   $( "#li.essenciais" ).addClass("fade-in");
82
83
84
85   $( "#li.menu.essenciais a" ).addClass("active text-primary");
86
87
88   if (screen.width < 768 || x < 768) {
89
90     $( "#.navbar-header .feedback" ).removeClass("hidden");
91     $( "#.navbar-header .feedback" ).addClass("fade-in");
92
93     $( "#.navbar-header .myZoom" ).addClass("hidden");
94
95     $( "#.navbar-header .valZoom" ).addClass("hidden");
96
97   } else {
98
99     $( "#nav li.feedback" ).removeClass("hidden");
100     $( "#nav li.feedback" ).addClass("fade-in");
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127

```

Figura 56 - Módulo GUI: Código do Menu

Na segunda fase, através de uma classe é definido a forma de execução da animação, como forma de movimento, neste acaso através de *ease-in*, que indica que arranca de forma lenta e aumenta de forma gradual a sua velocidade. Também são definidas as propriedades de duração bem como o *delay* (atraso) do arranque da animação.

Como mencionado na descrição geral deste módulo no num dos subcapítulos do capítulo Protótipo, onde mencionava que ao contrário do SVG que adapta – se a qualquer dispositivo, no elemento Canvas é necessário verificar as medidas dos dispositivos, para aparecer de forma correta ao utilizador, onde o código desenvolvido para esse efeito encontra-se na figura abaixo:

```
425 // adapt canvas to size of screen
426 function Mscreen() {
427
428
429
430     if (screen.width < 330 || x < 330)
431     {
432         S("#cvs").attr("width", 280);
433         S("#cvs").attr("height", 400);
434     }
435
436     else if (screen.width < 362 || x < 362)
437     {
438         S("#cvs").attr("width", 330);
439         S("#cvs").attr("height", 420);
440     }
441
442     else if (screen.width < 482 || x < 482)
443     {
444         S("#cvs").attr("width", 440);
445         S("#cvs").attr("height", 220);
446     }
447
448     else if (screen.width < 642 || x < 642)
449     {
450         S("#cvs").attr("width", 500);
451         S("#cvs").attr("height", 250);
452     }
453
454
455     else if (screen.width < 990 || x < 990)
456     {
457         S("#cvs").attr("width", 720);
458         S("#cvs").attr("height", 320);
459     }
460
461     else {
462
463         S("#cvs").attr("width", 800);
464         S("#cvs").attr("height", 400);
465     }
466 }
467
468
469 }
470
```

Figura 57 - Módulo GUI: Código Adaptação das dimensões do Canvas

Sendo esta uma das lacunas que infelizmente ao este elemento, onde tem que verificar a certas medidas de ecrã para aplicar a medida mais correta para o ecrã que utiliza naquele momento.

Em termos de suporte de Browsers, os mais afetados, poderão ser utilizadores do Browser IE, mais especificamente só a partir da versão 9, existe suporte a *media queries*. Relativamente a animação que é executada entre menus no Browsers IE, só existe suporte partir de IE10.

Neste módulo como melhoria futura passará por o utilizador alterar as cores a volta da interface gráfica, desde fundo do menu de branco a preto por exemplo, à cor do fundo que dentro quer por fora do elemento Canvas, acrescentando um sub-módulo chamado de temas.

3.3.2 Módulo Ficheiros

Neste módulo envolve carregamento de imagens quer através do disco do utilizador, imagem já contida no servidor (sample / amostra) ou até mesmo pela captação por uma Webcam.

Neste seção são detalhadas duas formas de carregamento, uma forma tradicional existente neste tipo de Software e outra maneira pouca usual ainda, dentro daqueles Softwares que foram apresentados no capítulo Estado de Arte.

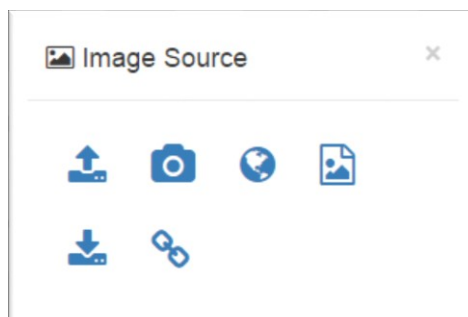


Figura 58 - Módulo Ficheiros: Menu Image

A primeira forma diz respeito a seleção de uma imagem ao disco do utilizador, onde o primeiro ícone diz respeito ao carregamento da imagem que esta no disco do utilizador, que antes de colocar a imagem no elemento Canvas pode pré-visualizar a imagem.

Para o desenvolvimento desta funcionalidade foi utilizada a API File Reader, conforme se pode visualizar na figura abaixo:

```

this.upload = function (file, preview) {
    var f = document.getElementById(file).files[0]; // get only one file or the
    first file

    var fr = new FileReader();

    fr.onload = function (e) {
        $(preview).attr("src", e.target.result);
        $(preview).removeClass("hidden");
        pic.src = $(preview).attr("src");
    };

    fr.readAsDataURL(f);

    return true;
};

```

Figura 59 - Módulo Imagem: Código Upload Imagem do Disco

Após a seleção da imagem contida no disco, a imagem é colocado temporariamente num elemento de imagem. Após a confirmação da imagem via pré-visualização, é colocado ou desenhado no elemento Canvas. Guardando uma cópia da imagem original noutra elemento de imagem.

Na figura abaixo encontra-se a segunda forma menos tradicional, a nível de dispositivos não móveis, mas que pode utilizar caso tenha uma Webcam integrada e externa.

Esta funcionalidade utiliza a API `getUserMedia`, conforme se pode ver a sua utilização de captação de imagem, na figura abaixo:

```
var webCam = {
  // Get the video element.
  video: document.querySelector(options.video),
  // Get the canvas element.
  canvas: document.querySelector(options.canvas),
  // Get the canvas context.
  ctx: canvas.getContext('2d'),
  // Get the capture button.
  captureBtn: document.querySelector(options.captureBtn),
  // This will hold the video stream.
  localMediaStream: null,
  // This will hold the screenshot base 64 data url.
  dataURL: null,
  // This will hold the converted PNG url.
  imageURL: null,
  // Get the input field to paste in the imageURL.
  imageURLInput: document.querySelector(options.imageURLInput),

  initialize: function () { ... },

  // Stream error.
  noStream: function (err) {
    alert('Could not get camera stream. ');
    console.log('Error: ', err);
  },

  // Stream success.
  gotStream: function (stream) { ... },
  // Capture frame from live video stream.
  capture: function () { ... },

  cancelStop: function () { ... }

};
```

Figura 60 - Módulo Imagem: Código Captação de Imagem Webcam

Para inicial a captação de imagem via Webcam é necessário, por parte do Browser, pedido de autorização ao utilizador. Após essa autorização, é colocada dentro de um elemento de vídeo temporariamente. Após um pressionar noutro botão, é captado a imagem para um elemento Canvas provisório. Posteriormente à captação, a câmara é desligada, podendo o utilizador fazer três ações. Sendo a primeira aprovar a imagem, que será utilizada para ser editada. Voltar a ligar a camara para tirar captar outra imagem, ou cancelar a captação e voltar ao menu inicial.

Ao longo do desenvolvimento foram surgindo dificuldades, mais precisamente na funcionalidade *upload* (carregamento) da imagem por URL (*Uniform Resource Locato*). Que devido às restrições de implementar a nova API CORS (*Cross-Origin Resource Sharing*), por implicar colocar a permissão de CORS do lado servidor onde está a imagem pretendida. E não sabendo se existe ou não essa autorização no respetivo ficheiro *htaccess* (*Hypertext access*). Como alternativa, foi utilizada a linguagem de script PHP, onde através de AJAX, é enviado o respetivo URL, para o script para ser tratada a imagem como se estivesse do lado da aplicação.

Também é de salientar dos Browsers da Microsoft, somente o Edge, é capaz de captar imagem através da Webcam, que infelizmente mesmo a ultima versão do IE. O IE11 não é capaz de utilizar esta API incluída na especificação do HTML5. Bem como tanto os Browsers da Microsoft como da Apple não suportam o novo atributo *download*, disponível num elemento *<a>*. Para contornar o problema foi colocado, este Browsers, abrir a imagem numa nova janela com a respetiva janela, fazendo utilizador guardar como através do rato do lado direito para salvar a imagem.

Para melhorias futuras passará por reduzir de forma que seja mais perceptível a criação do URL da imagem a ser partilha. Passará também a partilha em redes sociais como Facebook ou LinkedIn, bem como o possível carregamento de imagem que estejam na nuvem como Dropbox ou MeoCloud. Por último é necessário selecionar o problema a quando se carrega uma nova imagem, que fica sempre a anterior, onde para já, é necessário atualizar ou fazer F5 para forçar limpar o Canvas antes de introduzir uma nova imagem.

3.3.3 Módulo Essenciais

Neste módulo encontram-se inseridas a funcionalidades como, Crop, rotação, Fip (horizontal e vertical), Zoom e Pan (movimentar a imagem com o rato).

Como indicado no subcapítulo Módulos do Protótipo, foi um dos módulos mais complexos e onde foi utilizado mais tempo a desenvolver devido às funcionalidades como *Zoom* e *Crop*.

Na implementação da funcionalidade *Crop*, que exigiu despender mais tempo em pesquisa e desenvolvimento de que maioria da funcionalidade existentes, que juntamente jQuery, mais precisamente jQuery UI, foi possível em conjunto com Canvas API implementar esta funcionalidade, estando na categoria de mínimas / obrigatórias na aplicação.

```
3 ▶ $("#crp").click(function () {...});
4
5 ▼ $("#cropping").click(function () {
6
7     console.log("entrei");
8     var y = $("#overlay").css("top");
9     var x = $("#overlay").css("left");
10    var width = $("#overlay").css("width");
11    var height = $("#overlay").css("height");
12    console.log(x + " : " + y + " : " + width + " : " + height);
13    croproctangle(parseInt(x.replace('px', '')),
14                  parseInt(y.replace('px', '')), parseInt(width.replace('px', '')), parseInt(height.replace('px', '')));
15
16 });
17
18 ▼ function croproctangle(x, y, width, height) {
19     var tmp = document.createElement('canvas'),
20         ctx = tmp.getContext('2d');
21
22
23     console.log("x: " + x + " Y: " + y + " W: " + width + " H: " + height + " : " + parseInt(-250 + x) + " : " + parseInt(-250 + y));
24     var imageData = element.getImageData(x - 70, y - 75, width, height); //NOT translated corodinales X and Y only
25     ctx.canvas.width = width;
26     ctx.canvas.height = height;
27     ctx.putImageData(imageData, 0, 0);
28
29     $("#prevModal").attr("src", tmp.toDataURL());
30
31 }
32
33 ▼ $("#confirmCrop").click(function () {
34
35     element.clearRect(-upimg.width / 2, -upimg.height / 2, upimg.width, upimg.height);
36
37     //get image until terminate
38     upimg.onload = function () {
39
40         element.drawImage(upimg, -upimg.width / 2, -upimg.height / 2, upimg.width, upimg.height);
41     };
42
43     upimg.src = $("#prevModal").attr("src"); // source of image
44
45 });
46
47 $("#overlay").resizable({handles: "n, e, s, w, se, sw, nw, ne", containment: "#cvs"});
48 $("#overlay").draggable({containment: "#cvs"});
49
50 }
```

Figura 61 - Módulo Essenciais: Crop

Que através da jQuery UI, criou-se um componente que passará por cima da imagem. Que não poderá ultrapassar a área do Canvas. Esse elemento, com o id *overlay* como está na figura acima indicado, é responsável por extrair os valores da área selecionada. Após a seleção, tudo que esteja fora do retângulo, será eliminado com clique no botão / ícone a verde. Em suma, a dificuldade está, a semelhança de editores como Photoshop, haver componente de seleção, que neste caso é um retângulo (Figura 58), visível ao utilizar. Que pode-se arrastar, redimensionar e posteriormente extrair os valores dessa área para a função **croproctangle()**.

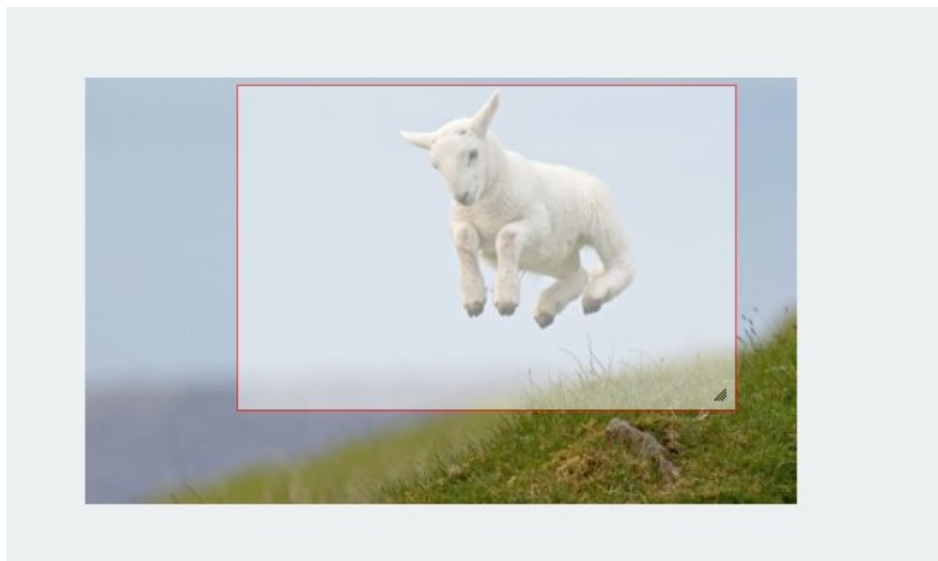


Figura 62 - Módulo Essenciais: Seleção Crop

Por último, mesmo a não estar incluído na categoria de mínimos, a funcionalidade Zoom, após análise do lançamento da primeira versão da aplicação, houve a necessidade de restringir o número de pixéis a ser apresentados a quando da visualização da imagem quer num dispositivo desktop quer num dispositivo móvel, onde foi necessário alterar que a nível de *design* quer a nível de programação quer a nível de interface gráfica.

Na figura abaixo, encontra-se código necessário tanto para implementar a funcionalidade de Zoom como Pan. Como indicado, a funcionalidade Zoom devido a nível performance, que poderia afetar, principalmente os dispositivos de menores dimensões. Sendo a funcionalidade Zoom acrescentada, devido a nível de performance, principalmente em dispositivos de reduzidas dimensões. Que era insuportável processar imagens de alta e até médias dimensões. O método por minimizar essa dificuldade, é o método **drawImage()**, que inicialmente coloca um valor predefinido a zoom a aplicar. E a imagem a desenhar no elemento Canvas aparecerá com metade das dimensões da imagem original. A imagem original com as respetivas dimensões é armazenada num elemento de imagem não visível. Onde as operações a serem aplicadas, primeiramente são executadas no elemento Canvas visível. Caso sejam para aplicar em definitivo, é enviada a imagem original para um Canvas temporário, com as operações a ser processadas em background.

```

280
281 v function drawImage() {
282
283     clear();
284
285     element.save();
286
287     element.scale(currentScale, currentScale);
288
289     element.drawImage(upimg, -upimg.width / 2, -upimg.height / 2, upimg.width, upimg.height);
290
291     element.restore();
292
293     console.log("drawing");
294 }
295
296 v $(".myZoom").on('input', function () {
297
298     currentScale = $(this).val();
299     console.log(currentScale);
300     $(".valZoom span").text(" " + currentScale);
301     drawImage();
302 });
303
304 // moviments of mouse | PAN
305 v canvas.onmousedown = function (e) {
306     var pos = getMousePos(canvas, e);
307     startX = pos.x;
308     startY = pos.y;
309     isDown = true;
310     console.log("x:" + startX + "-" + "y" + startY);
311
312 };
313
314 v canvas.onmousemove = function (e) {
315
316     var pos = getMousePos(canvas, e);
317
318     // colocado para apagar a imagem anterior
319 v if (isDown) {
320
321         var x = pos.x;
322         var y = pos.y;
323         clear();
324         element.translate(x - startX, y - startY);
325
326         drawImage();
327
328         startX = x;
329         startY = y;
330
331     }
332 };
333
334 v canvas.onmouseup = function (e) {
335     isDown = false;
336 };
337
338 v function getMousePos(canvas, evt) {
339     var rect = canvas.getBoundingClientRect();
340 v return {
341         x: evt.clientX - rect.left,
342         y: evt.clientY - rect.top
343     };
344
345 };

```

Figura 63 - Módulo Essenciais: Zoom e Pan

No que toca ás dificuldades do zoom, conforme se pode visualizar na figura acima no método **drawImage()**. Era necessário manter o estado existente da imagem bem como limpar corretamente o Canvas para aparecer no estado atualizado já com o Zoom desejado. Para seleccionar a primeira parte do problema, após pesquisa e estudo, foi encontrado no Canvas API, as seguintes funções: **element.save()**, **element.restore()** e **element.scale()**. Onde as 3 funções mencionadas, tem-se que colocar na ordem correcta, conforme se pode visualizar na figura acima. Em primeiro, **element.save()**, responsável por gravar em memória estado atual da imagem. Posteriormente, o utilizador ao mexer no componente responsável pode definir a quantidade. A função **element.scale()**, recebe dois parâmetros, um associado para escalar do eixo do horizontal e o segundo ao eixo da vertical. Onde esse valor da quantidade é igual tanto na horizontal como na vertical, para manter a proporcionalidade da imagem. Finalmente com **element.restore()**, volta ao estado inicial da imagem já com a proporção definida pelo o utilizador. Mas somente com o processo anterior não era suficiente porque a imagem não aparecia corretamente, conforme-se se pode visualizar na imagem seguinte:

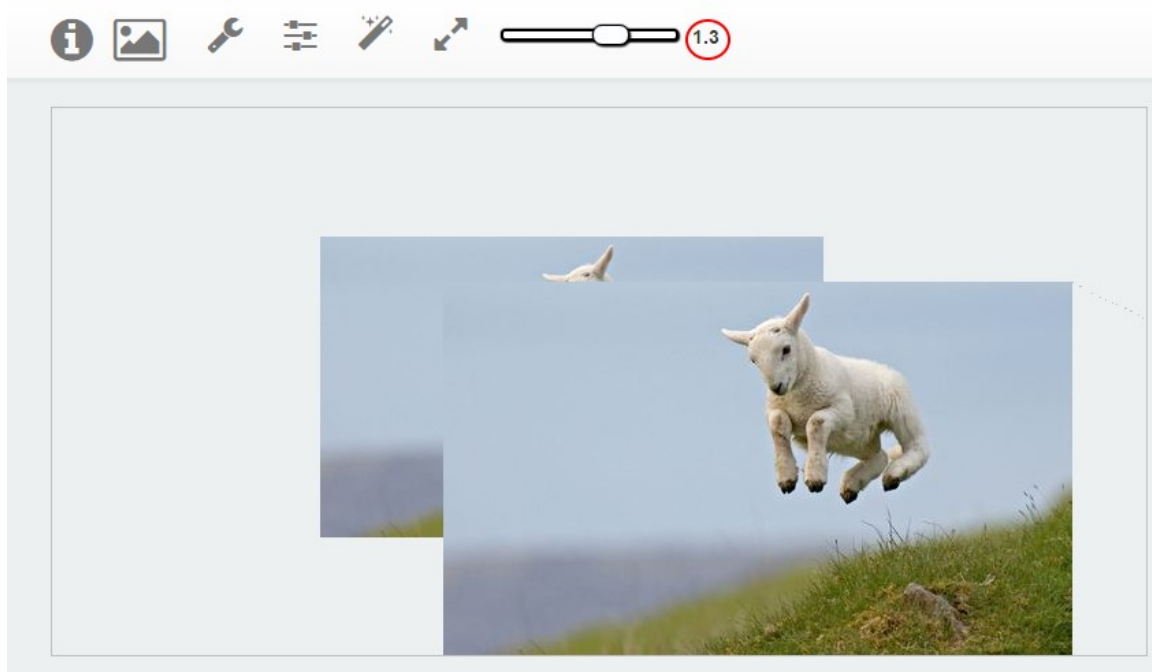


Figura 64 - Módulo Essenciais: Erro em Zoom

O que a figura acima demonstra, é após redesenhar no Canvas mantinha que o estado anterior e o estado atual. Para selecionar a segunda parte do problema, era preciso forçar o limpar o Canvas. Mas simplesmente limpar o Canvas com a função `element.clearRect()`, com os valores iguais em `element.drawImage()` conforme está na figura com o código do Zoom. Foi necessário acrescentar o valor da quantidade do Zoom.

No que toca a Browsers, no Microsoft IE11 as funcionalidades de Zoom e a quando de *Download* não funcionam, sendo forçado o Browser a abrir uma nova janela com a respetiva imagem para depois poder fazer o *Download*. Onde no Browser mais recente, Microsoft Edge, funciona bem a nível de Zoom, mantendo-se o problema em nível de *Download*. No Browser Safari, sucede a mesma coisa no que toca a *Download* da imagem.

Quanto a melhorias, é necessário transpor o resultado final do *Crop* para a versão original da imagem bem como selecionar o problema após carregar a imagem se o utilizar for arrastar a imagem, sem antes for utilizado o Zoom, o Pan (arrastar) não funcionará corretamente.

3.3.4 Módulo Ajustamentos

Neste módulo referente a implementação de filtros, com o intuito de melhorar e degradar a imagem de forma controlada. Para aplicar os filtros os utilizador tem de selecionar o ícone de adjustments. Selecionar o respetivo filtro com a quantidade desejada, conforme se pode visualizar na figura abaixo:

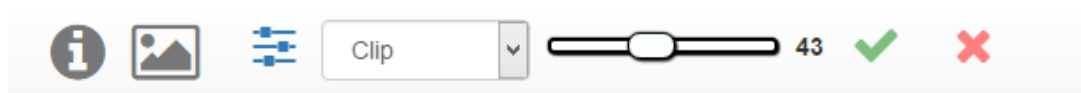


Figura 65 - Módulo Ajustamentos: Menu

Onde *Clip*, é nome do filtro selecionado, o numero 43 diz respeito à quantidade aplicar na imagem. À semelhança dos restantes menus o ícone verde serve para aplicar o filtro e volta para o menu inicial. E o ícone vermelho cancela o filtro, voltando para o estado anterior bem como para o menu inicial.

Neste módulo, para a sua implementação é utilizada a Frameworks CamanJS, conforme se pode verificar na figura seguinte:

```
213 // execute the selected filter
214 function applyFilter(inp, cvs,inter) {
215
216     var caman = Caman(cvs), x;
217
218     $(".res span").html(x);
219
220     switch (inp) {
221
222         case "bri":
223
224             $(".#bri").change(function () {
225
226                 x = $(".#bri").val();
227
228                 $(".res span").html(x);
229
230                 caman.revert();
231
232                 caman.brightness(x);
233
234                 if (inter !== "") {
235                     caman.render(function () {
236
237                         $(".inter").attr("src", caman.toBase64());
238
239                         console.log("inter applied");
240
241                     });
242                 }
243             });
244
245         else{
246             caman.render();
247             console.log("no inter");
248         }
249     });
250     break;
251
252     case "blu":
253         $(".#blu").change(function () { ... });
254         break;
255
256     case "cli":
257         $(".#cli").change(function () { ... });
258         break;
259
260     }
261 }
```

Figura 66 - Módulo Ajustamentos: Código

Na figura acima encontra-se a implementação do método applyFilter, tem como argumentos o filtro selecionado pelo utilizador, enquanto segundo diz respeito a qual elemento Canvas aplicar o filtro. O último argumento é opcional, onde possibilita indicar um segundo elemento Canvas onde será guardada a versão final a aplicar.

Ao todo foram implementados cerca de 12 filtros, sendo estes apresentados no exemplo existe no Website da Framework, que são seguintes:

- Brightness (luminosidade)
- Blur (Desfocar)
- Clip
- Contrast (Contraste)
- Exposure
- Gamma
- Hue
- Noise (Ruido)
- Saturation
- Sepia
- Sharpen
- Vibrance

Ao longo do desenvolvimento deste módulo encontraram dificuldades, que posteriormente foram ultrapassadas, lendo documentação fornecida pelo Website da Frameworks e alguma pesquisa em outros Websites em relação ao respetivo tópico, como fazer voltar à imagem inicial bem como a partir da Frameworks converter para imagem.

Como melhorias futuras, a inserção de mais filtros mais personalizáveis quer através de RGB quer através manipulação do Kernel, método fornecido pela Frameworks.

3.3.5 Módulo Efeitos

Módulo efeitos à semelhança do módulo anterior, também utiliza é utilizada a Frameworks CamanJS para implementa efeitos. Que tem como objetivo possibilitar ao utilizador alterar a imagem de forma drástica e rapidamente, em contraste o que acontece no módulo anterior.

```
639 function applyEffect(cvs, inter) {
640     var caman = Caman(cvs);
641     var se;
642     // get id of btn and respective effect
643
644
645
646     $("#selEffect").change(function () {
647         se = $("#selEffect option:selected").val();
648         console.log(se);
649         switch (se) {
650             case "ef1":
651                 caman.revert();
652                 caman.vintage();
653                 if (inter !== "") { ... } else {
654                     caman.render();
655                     console.log("no inter");
656                 }
657                 // toUrl(inter);
658                 console.log("ok");
659                 break;
660             case "ef2":
661                 caman.revert();
662                 caman.lomo();
663                 if (inter !== "") { ... } else {
664                     caman.render();
665                     console.log("no inter");
666                 }
667                 break;
668         }
669     });
670 }
```

Figura 67 - Módulo Efeitos: Código

Na figura 44 pode visualizar um fragmento da implementação deste módulo. Onde o método applyEffect recebe dois argumentos, onde o primeiro diz respeito ao destino onde será aplicado o efeito, neste caso aponta um elemento Canvas, enquanto o segundo referente a indicação para onde vai, caso seja necessário, o resultado do filtro em formato imagem. Conforme o efeito selecionado pelo utilizador, é verificado o evento de mudança de seleção na lista de efeitos, em caso de mudança é aplicado o respetivo efeito selecionado.

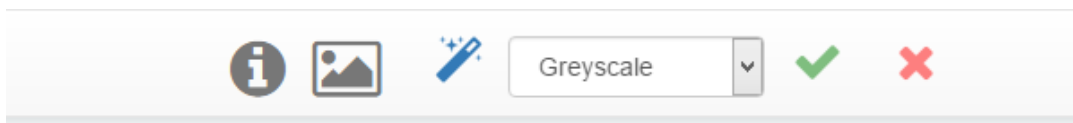


Figura 68 - Módulo Efeitos: Menu

Onde ao todo foram implementado cerca de 19 efeitos, com os seguintes nome, conforme o exemplo também existente com os respetivos nomes:

- Vintage
- Lomo
- Clarity
- Sin City
- Cross Process
- Orange Peel
- Love
- Grungy
- Jarques
- Pinhole
- Old boot
- Glowing Sun
- Sunrise
- Grayscale
- Magesty
- Nostalgia
- OldBoot

No que toca a dificuldades foram semelhante, vistos ter sido desenvolvido com a mesma Frameworks do módulo anterior e ultrapassadas de forma semelhante.

Em termos de melhorias futuras, é de mencionar que pode passar por o acrescentar de mais, através personalização, visto que os implementado serem um agrupamento de filtros com valores de quantidades fixas, criar também novos efeitos através da conjugação de outros filtros com valores de quantidade que assentem bem.

3.3.6 Módulo Texto

Como já num subcapítulo antes referente a este módulo, o elemento Canvas e a respetiva API não existe grande suporte ou evolução para implementar funcionalidades como Cursor, multi-linha, no caso de se mudar de linha, entre outras funcionalidades que possibilita aos programadores implementar, á semelhança existir em editores de texto.

Na interface gráfica o utilizador ao selecionar botão o ícone à semelhança de uma caneta, aparecerá uma janela com as propriedades do texto, figura 45. Onde pode modificar antes de começar a escrever na tela, mais concretamente, em cima do elemento Canvas. Após ter selecionado as propriedades desejadas, ao clicar no elemento Canvas aparecerá um cursor onde o utilizador poderá começar a escrever o respetivo texto. Posteriormente, ao selecionar outra funcionalidade, o respetivo cursor desaparecerá.

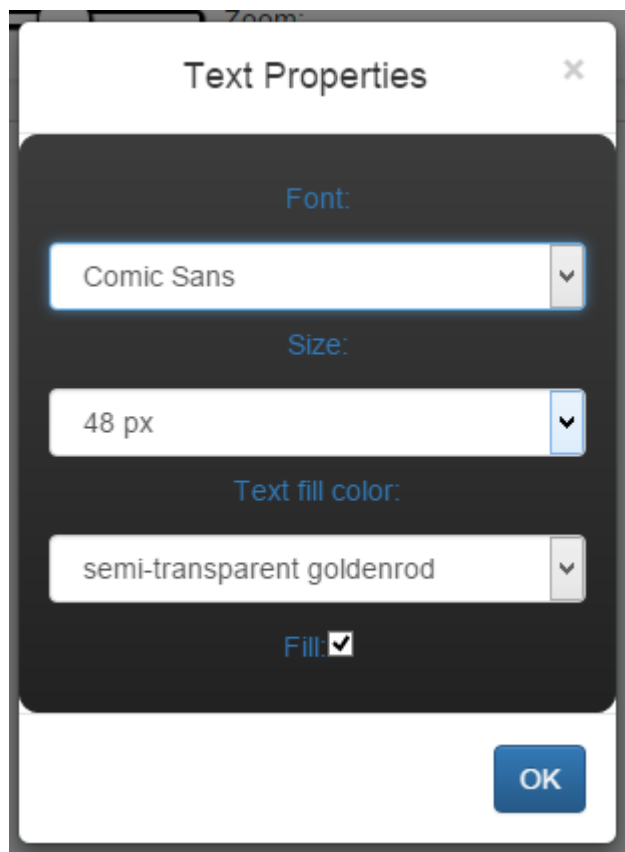


Figura 69 - Módulo Texto: Propriedade do Texto

Na figura abaixo o código mais relevante, sendo o núcleo deste módulo, baseado no exemplo existe no livro Core HTML5 Canvas.

```
33 // Cursor.....
34
35 ▼ TextCursor = function (fillStyle, width) {
36     this.fillStyle = fillStyle || 'rgba(0, 0, 0, 0.7)';
37     this.width = width || 2;
38     this.left = 0;
39     this.top = 0;
40 };
41
42 ▼ TextCursor.prototype = {
43 ▼   getHeight: function (context) {
44       var h = context.measureText('M').width;
45       return h + h / 6;
46   },
47
48 ▶   createPath: function (context) { ... },
49
50 ▶   draw: function (context, left, bottom) { ... },
51
52 ▶   erase: function (context, imageData) { ... }
53 };
54
55 // Text lines.....
56
57 ▼ TextLine = function (x, y) {
58     this.text = '';
59     var cv=$("#cvs").get(0);
60     this.left = x-cv.width/2;
61     this.bottom = y-cv.height/2.;
62     this.caret = 0;
63 };
64
65 ▼ TextLine.prototype = {
66 ▼   insert: function (text) {
67       this.text = this.text.substr(0, this.caret) + text +
68       this.text.substr(this.caret);
69       this.caret += text.length;
70   },
71
72 ▶   removeCharacterBeforeCaret: function () { ... },
73
74 ▶   getWidth: function (context) { ... },
75
76 ▶   getHeight: function (context) { ... },
77
78 ▶   draw: function (context) { ... },
79
80 ▼   erase: function (context, imageData) {
81       context.putImageData(imageData, 0, 0);
82   }
83 };
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
```

Figura 70 - Módulo Texto: código do Core

Encontra-se, na figura acima, os métodos referentes a funcionalidades como o cursor bem como o aparecimento de texto numa linha bem como a possibilidade de utilizador remover os caracteres inseridos nessa linha enquanto estiver a escrever.

Através do código anterior foi necessário criar os respetivos métodos para o evento click do rato e respetivas coordenadas bem como a possibilidade de alteração das propriedades do texto antes de escrever no Canvas como tamanho de texto, fonte de texto e respetiva inserção no elemento Canvas.

Como referido anteriormente, a funcionalidade está na categoria opcional devido ao grau de dificuldade da sua implementação, sendo uma das limitações da tecnologia HTML5, mas concretamente HTML5 Canvas e respetiva API, no que toca a texto ser muito básica onde o tanto o grupo designado pela W3C e WHATWG não deram grande importância a este tipo de funcionalidade.

Como possíveis melhorias ou alterações, devido à limitações anteriormente mencionadas, passa alterar a maneira como utilizador interage com esta funcionalidade, introduzindo uma caixa de texto onde o utilizador escreveria o respetivo texto, em vez de escrever diretamente o texto no elemento Canvas, após o evento clique no elemento Canvas, obtendo as respetivas coordenadas.

3.3.7 Módulo Ecrã inteiro

Este o módulo com menos linhas de código comparando, com os restantes, onde o utilizador quer através do ícone de fullscreen que através de teclas de atalho Enter ou F11 (no caso de sistema operativo Windows), é possível mostrar somente, escondendo tudo que seja referente ao Browser utilizado, onde após se entrar em modo fullscreen, o browser dá a indicação da tecla de atalho para poder sair do mesmo, sendo elas as mesmas anteriormente referidas bem como ESC.

Na figura abaixo encontra código implemento referente a este módulo:

```
509
510 ▼ function toggleFullscreen() {
511     if (!document.fullscreenElement && // alternative standard method
512         !document.mozFullScreenElement && !document.webkitFullscreenElement && !document.msFullscreenElement)
513     { // current working methods
514
515         if (document.documentElement.requestFullscreen) {
516             document.documentElement.requestFullscreen();
517         } else if (document.documentElement.msRequestFullscreen) {
518             document.documentElement.msRequestFullscreen();
519         } else if (document.documentElement.mozRequestFullScreen) {
520             document.documentElement.mozRequestFullScreen();
521         } else if (document.documentElement.webkitRequestFullscreen) {
522             document.documentElement.webkitRequestFullscreen(Element.ALLOW_KEYBOARD_INPUT);
523         }
524
525         // change the icon of fullscreen to icon associate of exit of fullscreen
526         $("#full i").removeClass("fa-expand");
527         $("#full i").addClass("fa-compress");
528
529     }
530
531     else {
532         if (document.exitFullscreen) {
533             document.exitFullscreen();
534         } else if (document.msExitFullscreen) {
535             document.msExitFullscreen();
536         } else if (document.mozCancelFullScreen) {
537             document.mozCancelFullScreen();
538         } else if (document.webkitExitFullscreen) {
539             document.webkitExitFullscreen();
540         }
541
542         // remove the icon of exit of fullscreen to restore icon of fullscreen
543         $("#full i").removeClass("fa-compress");
544         $("#full i").addClass("fa-expand");
545     }
546 }
```

Figura 71 - Módulo Fullscreen: Código

Onde na função *toggleFullscreen*, primeiramente verifica se atualmente o Browser está em modo de ecrã inteiro, caso não esteja é executado o pedido para aplicar o modo associado com o consequente ícone a ser mudado, caso já esteja, é efetuado o pedido para sair do modo de ecrã, com o icon de fullscreen, semelhante ao elemento de vídeo.

É de referir que, no código anterior é feito a verificação qual o browser devido, a alguma versões mais antigas dos browser ainda não ter de raiz o método padrão *document.fullscreenElement*.

3.4 Conclusão

Neste capítulo são abordados tópicos acerca do projeto proposto. Que consiste no desenvolvimento de uma aplicação Web-based, com objetivo de manipulação e edição de imagem. Onde o foco principal é a tecnologia HTML5, mais precisamente a utilização do elemento Canvas e a respetiva API. Em conjunto com outras tecnologias Web como CSS3 e JavaScript.

Para o desenvolvimento da interface gráfica do utilizador, esteve de base as aplicações ou Software apresentadas no capítulo “Estado da Arte” bem como tecnologias e Frameworks para impulsionar o respetivo desenvolvimento.

Foram também indicadas neste capítulo, as tecnologias e Frameworks ou bibliotecas apresentadas no capítulo 2, com que foi desenvolvido o protótipo demonstrado.

Sendo que no subcapítulo Protótipo, além indicação das tecnologias aplicadas e Framework ou biblioteca utilizadas, foram descritas os módulos que consiste a aplicação. Com respetivos funcionalidades desenvolvidas.

Para além disso o protótipo é sujeito a uma avaliação parte de um conjunto de alunos do ISEP, que foi dividida em dois inquéritos, cujos dados dos inquéritos permitiu melhorar o processo de desenvolvimento bem como fazer melhorias tanto ao nível visual como a nível de funcionalidades.

No que toca a implementação em concreto, o protótipo está organizado 7 módulos, cada um com objetivo específico e com a respetivas funcionalidades associadas.

Para o seu desenvolvimento foram utilizadas tecnologias web como HTML5 e algumas das suas APIs. Bem como CSS3 e JavaScript, sendo utilizadas Frameworks o bibliotecas para um desenvolvimento mais eficiente.

A maioria das funcionalidades implementadas com as tecnologias referidas são suportadas pela maior parte dos Browsers mais utilizados como Google Chrome, Mozilla Firefox, Opera, Safari e IE.

Contudo os dois últimos Browsers em termos de suporte de algumas das funcionalidades incluídas tanto em HTML5 como CSS3 estão muito abaixo ou atrasados em comparação aos restantes.

Sendo que no Safari não suporta funcionalidade de acesso à Webcam bem como não reconhece o atributo *download* ao pressionar no elemento `<a>` para descarregar a imagem.

Quanto aos Browsers da Microsoft, o suporte é ainda mais fraco. Que para além terem os mesmos defeitos do safari, em versões igual ou abaixo do 9 não suporta funcionalidade do CSS3, como *media queries* e animações ou transições através de CSS. Bem como a funcionalidade

Zoom não funciona nos Browsers IE. Mas o novo Browser da Microsoft, Edge ao contrário das versões IE, suporta a API HTML5 de acesso a WebCam.

No que toca a implementação das funcionalidades propostas, todas que estavam catalogadas como prioritárias estão incluídas no protótipo. E outras que eram complementares como Zoom, acesso Webcam.

Das funcionalidades implementadas, as que deram mais dificuldades foram *Crop*, que somente funciona em dispositivos desktop. O Zoom, que motivos de performance foi necessário implementar. E por último texto, permitindo de forma básica alterar algumas das propriedades do texto.

A tecnologia CSS3 está mais estável nos principais Browsers, tendo maior suporte, nas funcionalidades utilizadas no protótipo.

Sendo o foco neste documento a tecnologia HTML5, mais concretamente o elemento Canvas para a edição e manipulação da imagem, a maior limitação no HTML5 Canvas recai sobre a sua API para texto que é limitada.

Por último é de referir que, ao longo do tempo de pesquisa e desenvolvimento, o HTML5 Canvas é associado muitas vezes para desenvolvimento de jogos. Ou em que muito das bibliografias em que foi feito o estudo somente menciona Canvas, para além do contexto de desenvolvimento de jogos, de criação de aplicações para desenho de figuras básicas geométricas. Referido muito pouco no que toca a edição e manipulação de imagens, mais concretamente de pixels.

4 Conclusão

4.1 Resumo e objetivos alcançados

Neste documento foi apresentado a implementação de uma aplicação web-based, proposta no âmbito da Tese de Mestrado de Engenharia Informática, ramo de Sistemas Gráficos e Multimédia, no ISEP (Instituto Superior Engenharia do Porto). Com objetivo de edição e manipulação de imagens, tanto em desktop como dispositivos móveis. Este processo é exclusivamente feito no lado do cliente, ou seja, no lado do Browser do utilizador. No lado do servidor, servirá somente para o armazenamento da aplicação e respetivos recursos.

Numa primeira fase foram estudados soluções dedicadas a edição e manipulação de imagem. Posteriormente foram feitas análises comparativas como a nível de plataformas suportadas e funcionalidades existentes. Cujo propósito é observar as várias interfaces gráficas, a que funcionalidades contém, como estão organizadas bem como qual a tecnologia utilizada.

Com base na observação e respetiva comparação entre programas selecionados, foi desenvolvido a interface gráfica bem como elaborada uma primeira listagem de possíveis funcionalidades a incorporar.

Posteriormente, foi feito um levantamento de tecnologias que fornecessem capacidades de desenvolver o projeto proposto. Esta investigação visou o estudo das capacidades das tecnologias web mais moderna, como HTML5, CSS3 e JavaScript, como sendo tecnologias alternativas à tecnologia Adobe Flash.

No desenvolvimento da interface gráfica do protótipo foi utilizada tecnologia CSS3, que através da framework Twitter bootstrap, foi possível colocar o protótipo acessível a qualquer dispositivo, qualquer que seja as suas dimensões. Das duas tecnologias que estão dentro do HTML5, foi escolhida Canvas e a respetiva API, que juntamente com JavaScript consegue de forma mais eficiente executar o propósito do protótipo, que é editar e manipular imagens, mais concretamente os respetivos pixéis.

Foram implementadas no protótipo todas as funcionalidades catalogadas como obrigatórias. As funcionalidades complementares de texto, Webcam, *Pan* e *Zoom* também foram implementadas com sucesso. A aplicação funciona plenamente em qualquer dos dispositivos alvo, com exceção das funcionalidades de crop e texto que estão otimizadas para dispositivos de média ou grande dimensão e que contenham rato e teclado. Alguns Browsers limitam as capacidades das aplicações web, e por isso um pequeno conjunto de funcionalidades não estão disponíveis em alguns Browsers. É o caso da captura de imagem via Webcam que não está operacional em Apple Safari e no Microsoft Internet Explorar.

A implementação deste protótipo permitiu ter uma visão mais abrangente das potencialidades e limitações das aplicações web hoje em dia. Com efeito concluiu-se que é possível implementar com recursos moderados uma aplicação completa de edição de imagem funcional e dirigida a qualquer utilizador comum da web. Mesmo assim ainda há alguns constrangimentos neste desenvolvimento que estão dependentes essencialmente de limitações de APIs implementadas pelos Browsers. Um exemplo destas limitações é a API do Canvas que não tem suporte adequado para edição de texto.

Outra limitação encontrada prende-se com sistemas de controlo de segurança que demarcam a interação entre um Browser e múltiplos servidores. Efetivamente por omissão uma aplicação web-based implementada num cliente Browser apenas tem acesso a recursos no servidor de onde foi descarregada, o que pode ser limitativo para aplicações como a apresentada. Esta situação pode ser ultrapassada usando o servidor como "proxy", mas a CORS API permite a ligação direta a outros servidores desde que o servidor seja configurado adequadamente.

Por último, os inquéritos que foram disponibilizados aos alunos de Mestrado de Engenharia informática do ISEP possibilitaram a avaliação e evolução do protótipo tanto a nível do processo de desenvolvimento em si, como a nível de melhorias de algumas das funcionalidades.

4.2 Trabalho Futuro

Como todas as aplicações, há grande margem de manobra para evolução, especialmente quando falamos de aplicações web cujas tecnologias de base estão em constante desenvolvimento. Isto é especialmente importante no caso de aplicações de edição e manipulação de imagens em que as APIs disponíveis são relativamente recentes e, como tal, sujeitas a ajustes e melhorias mais dramáticas.

Da mesma forma o protótipo apresentado também tem muito potencial de evolução, começando pelas falhas e possíveis melhorias assinalados nos inquéritos realizados. Além disto funcionalidades complementares como molduras e desenho vetorial poderiam melhorar significativamente as alternativas de criatividade na utilização da aplicação.

Por último, seria interessante disponibilizar a aplicação num modo offline. Em que na primeira vez que aplicação era executada online, os recursos ficariam em cache no browser. Sendo isto possível através da API App Cache, que se encontra na especificação do HTML5, tendo vantagens principalmente nos dispositivos móveis. Além de poupar dados móveis do pacote de dados de internet, possibilitaria a utilização das aplicações mesmo em situações sem ligação à rede.

Referências

- [Adam Roberts, 2014] Adam Roberts. "Introduction to HTML5 Canvas Element". <http://www.sitepoint.com/web-foundations/introduction-html5-canvas-element/> [último acesso: Setembro de 2015]
- [Anthony Godinho, 2014] Anthony Godinho. "Sumo Paint Review". <http://ajgodinho.hubpages.com/hub/sumo-paint-online> [último acesso: Setembro de 2015]
- [Autodesk Pixlr, 2015] Autodesk Pixlr. "Membership". <https://pixlr.com/memberships> [Último acesso: Setembro de 2015]
- [Brad Dayley, 2013] Brad Dayley. "jQuery e Javascript - PhraseBook". Addison-Wesley Professional, 2013. Pag.4-5
- [Brad Dayley (b), 2013] Brad Dayley. "jQuery e Javascript - PhraseBook". Addison-Wesley Professional, 2013. Pag.7
- [Brad Dayley (c), 2013] Brad Dayley. "jQuery e Javascript - PhraseBook". Addison-Wesley Professional, 2013. Pag.9
- [CamanJS, 2015] CamanJS. "Extending". <http://camanjs.com/guides/#Extending> [último acesso: Setembro de 2015]
- [CreateJS, 2015] CreateJS. "Intro". <http://www.createjs.com> [último acesso: Setembro de 2015]
- [Creative Adobe, 2014] Creative Adobe, *Creative Cloud Plans*, <https://creative.adobe.com/en/plans> [último Agosto de 2015].
- [Christina Warren, 2012] Christina Warren. "The Life, Death and Rebirth of Adobe Flash". <http://mashable.com/2012/11/19/history-of-flash/#kLTwXRqT45k4> [último acesso: Setembro de 2015]
- [Devicic, 2011] Ivana Devicic. "Mores From Picozu – The Online Image Editor". <http://teksocial.com/socialblog/2013/4/11/more-from-picozu-the-online-image-editor.html> [último acesso: Setembro de 2015]
- [Eisenberg et al, 2014] J. David Eisenberg e Amelia Bellamy-Royds. "SVG - Essentials". O'Reilly, segunda Edição, 2014, pág. 26-27
- [Eisenberg et al (b), 2014] J. David Eisenberg e Amelia Bellamy-Royds. "SVG - Essentials". O'Reilly, segunda Edição, 2014, pág. 20
- [Eisenberg et al (c), 2014]. J. David Eisenberg e Amelia Bellamy-Royds. "SVG - Essentials". O'Reilly, segunda Edição, 2014, pág. 21
- [Eliot Sykes, 2014] Eliot Sykes. "Touch gestoures with HammerJS". <http://www.webdevbreak.com/episodes/touch-gestures-hammerjs> [último acesso: Setembro de 2015]
- [Emily, 2015] Emily. "A brief History of Cascading Style Sheet". <http://howtocode.pk/css-tutorial/a-brief-history-of-cascading-style-sheets> [Último acesso: Setembro de 2015]
- [FabricJS, 2015] FabricJS. "Docs". <http://fabricjs.com/docs/> [Último acesso: Setembro de 2015]
- [FabricJS, 2015] FabricJS. "Deserialization, SVG parser". <http://fabricjs.com/fabric-intro-part-3/#deserialization> [último acesso: Setembro de 2015]
- [Frost et al, 2012] Jon Frost, David Dailey, Domenico Strazzullo. "Building Web Application With SVG". Microsoft. 2012. Pag.10
- [Flaton, 2013] Steven e Jeff Fulton. "HTML5 Canvas". O'reilly, 2013, 2ª Edição. <http://chimera.labs.oreilly.com/books/1234000001654/ch01.html> [Último acesso: Setembro de 2015]
- [GIMP, 2015] GIMP. "Feature Overview". <http://www.gimp.org/features/> [último acesso: Setembro de 2015]
- [HammerJS, 2015] HammerJS. "Getting Started". <http://hammerjs.github.io/getting-started/> [último Acesso: Setembro de 2015]

- [HTML5test, 2015], HTML5 Test. "HTML5 on Desktop Browsers". <https://html5test.com/results/desktop.html> [última Acesso: Setembro de 2015]
- [Keith et al, 2010] Jeremy Keith e Jeffrey Sambells. "DOM Script: Web Design with JavaScript and the Document Object Model". Apress, 2ª Edição, 2010. Pag.1.
- [Krishna et al.,2010] Krishna et al. "Implementation and Performance Evaluation of a Hybrid Distributed System for Storing and Processing Images from the Web". Cloud Computing Technology and Science (CloudCom), IEEE Second International Conference. Indianapolis, In. 2010. Disponível: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5708528&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5708528 [último Acesso: Agosto de 2015]
- [Goldstein et al, 2015] Alexis Goldstein, Louis Lazaris, Estelle Weyl. "HTML5 & CSS3 For The Real World, 2nd Edition", SitePoint, 2ª Edição, 2015. Pag.25.
- [Goldstein et al (b), 2015] Alexis Goldstein, Louis Lazaris, Estelle Weyl. "HTML5 & CSS3 For The Real World, 2nd Edition", SitePoint, 2ª Edição, 2015. Pag.19.
- [Hélder Ferreira, 2013] Hélder Ferreira. "Pixkr Express - O Photoshop da Autodesk para Android". <http://pplware.sapo.pt/smartphones-tablets/android/pixlr-express-o-photoshop-da-autodesk-para-android/> [Último acesso: Setembro de 2015]
- [Instagram, 2015] Intagram. "Instagram Stats", <https://instagram.com/press/> [último Acesso:07/09/2015]
- [iLex, 2011] iLex (blogdoiphone.com). "Steve Jobs tinha razão: adobe abandona o flash para dispositivos móveis". <http://blogdoiphone.com/2011/11/steve-jobs-tinha-razao-adobe-abandona-o-flash-para-dispositivos-moveis/> [último Acesso: Setembro de 2015].
- [Jack Dove, 2014] Jack Dove. "The best 9 Browser-based photo editing tools available today". <http://thenextweb.com/creativity/2014/02/24/9-browser-based-photo-editing-tools/> [último acesso: Setembro de 2015]
- [James Williams, 2011] James Williams. "Introductions to Raphael". http://www.html5rocks.com/en/tutorials/raphael/intro/?redirect_from_local=pt [Último acesso: Setembro de 2015]
- [JSter, 2015] JSter. "CamanJS". <http://jster.net/library/camanjs> [último acesso: Setembro de 2015]
- [Liang et al, 2009] Yi Liang et al. "An Efficient Image Processing Method Based on Web Services for Mobile Devices". Image and Signal Processing, 2009. CISP '09. 2nd International Congress. Tianjin. 2009. Disponível: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5302450&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5302450 [último Acesso: Setembro de 2015]
- [Jornal de Negócios, 2013] Jornal de Negócios. "Portugueses acedem cada vez mais à Internet através de dispositivos móveis". http://www.jornaldenegocios.pt/empresas/detalhe/portugueses_acedem_cada_vez_mais_a_internet_atraves_de_dispositivos_moveis.html [último Acesso: Setembro de 2015]
- [Mahesh Chand, 2012] Mahesh Chand. "Fabric.JS: A powerful JavaScript Library to work with Canvas". <http://www.c-sharpcorner.com/news/2260/fabric-js-a-powerful-javascript-library-to-work-with-canva.as> [último acesso: Setembro de 2015]
- [Manderscheid, 2014] Brad Manderscheid. "Beginning HTML5 Games with CreateJS". Apress, 2014. Pag.4
- [Manderscheid (b), 2014] Brad Manderscheid. "Beginning HTML5 Games with CreateJS". Apress, 2014. Pag.6
- [Manderscheid (c), 2014] Brad Manderscheid. "Beginning HTML5 Games with CreateJS". Apress, 2014. Pag.10

- [Manderscheid (d), 2014]
[MDN, 2015]
- Brad Manderscheid. "Beginning HTML5 Games with CreateJS". Apress, 2014. Pag.11
- MDN. "Getting Started with WebGL". https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL [último acesso: Setembro de 2015]
- [MDN (b), 2015]
- MDN, "Selectors". https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Selectors [último Acesso: Setembro 2015]
- MDN (c), 2015].
- MDN. "A Re-Introduction to JavaScript". https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript [último acesso: Setembro de 2015]
- [Michael Muchmore, 2014a]
- Michael Muchmore, *Adobe Photoshop CC 2014*, <http://www.pcmag.com/article2/0%2c2817%2c2420243%2c00.asp> /, página 2 [último Acesso: Agosto de 2015]
- [Michael Muchmore, 2014b]
- Michael Muchmore, *Adobe Photoshop CC 2014*, <http://www.pcmag.com/article2/0%2c2817%2c2420243%2c00.asp> /, página 1 [último Acesso: Agosto de 2015]
- [Microsoft, 2014]
- Microsoft. "SVG". [https://msdn.microsoft.com/library/hh673562\(v=vs.85\).aspx](https://msdn.microsoft.com/library/hh673562(v=vs.85).aspx) [último acesso: Setembro de 2015]
- [moock.org, 2014]
- Moock. "SVG – Features of Interest". <http://www.moock.org/webdesign/svg/articles/svg-featureset.html> [último Acesso: Setembro de 2015]
- [Mozilla, 2015]
- Mozilla. "ECMAScript 6 support in Mozilla". https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_6_support_in_Mozilla [último acesso: Setembro de 2015]
- [Myeongjin Cho et al, 2014].
- Myeongjin Cho et Al. "Web-based image processing using JavaScript and WebGL". Consumer Electronics (ICCE), 2014 IEEE International Conference. Las Vegas, NV. 2014. Disponível: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6776030&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel7%2F6761345%2F6775879%2F06776030.pdf%3Farnumber%3D6776030> [último Acesso: Setembro de 2015]
- [Nick Peers, 2015]
- Nick Peers. "Paint.Net 4 Final". <http://www.pcadvisor.co.uk/download/design-illustration/paintnet-406-final-3329486/> [último acesso: Setembro de 2015]
- [Norman, 2013]
- George Norman. "Fotor 2.0". <http://fotor.findmysoft.com> [último acesso: Setembro de 2015]
- [Patel, 2014]
- Sandeep Kumar Patel. "Developing Responsive Web Applications with Ajax and jQuery". Packt Publishing, 2014. Pag. 20
- [Pedro Pinto, 2015]
- Pedro Pinto. "Depois da Apple... Google também vai bloquear Flash". <http://pplware.sapo.pt/google/dep0is-da-applegoogle-tambm-vai-bloquear-flash/> [último acesso: Setembro de 2015]
- [Pedro Simões, 2014]
- Pedro Simões. "Pixlr, depois dos martphone chegou ao Window e o OSX". <http://pplware.sapo.pt/apple/mac/pixlr-depois-dos-smartphones-esta-app-chegou-ao-windows-o-osx/> [último Acesso: Setembro de 2015]
- [Pedro Simões (2), 2015]
- Pedro Simões. "Descobertas novas falhas no Flash usadas pela Hacking team". <http://pplware.sapo.pt/informacao/descobertas-novas-falhas-no-flash-usadas-pela-hacking-team/> [último acesso: Setembro de 2015]
- [Pedro Simões (3), 2015].
- Pedro Simões. "Adobe Atualiza Flash depois do bloqueio do Firefox". <http://pplware.sapo.pt/informacao/adobe-actualiza-flash-depois-do-bloqueio-do-firefox/> [último acesso: Setembro de 2015]
- [Peter Gasston, 2015].
- Peter Gasston, "The book of CSS3 - A developer's guide to the future of web design", No Starch Press, 2ª Edição, 2015, Pag. 2-3.
- [Qin Gang, 2014]
- Qin Gang et al. "A mobile Internet oriented system for multimedia data of nature reserve". Electronics, Computer and Applications, IEEE Workshop .

- Ottawa, ON. 2014. Disponível:
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6845734> [Último Acesso: Setembro 2015]
- [Kevin Lyn, 2015] Kevin Lyn. "Bootstrap 3.3: A guide for beginners to Learn Web Framework Fast". Amazon (kindle Version), 2015.
- [Raphael, 2015] Raphael. "Raphael JavaScript Library". <http://raphaeljs.com> [Último Acesso: Setembro de 2015]
- [Rauschmayer, 2014] Axel Rauschmayer. "Speaking JavaScript". O'Reilly Media, 2014. Disponível: <http://speakingjs.com/es5/ch05.html> [último acesso: Setembro de 2015]
- Rob Crowther et al, 2014] Rob Crowther, Joe Lennon, Ash Blue, Greg Wanish. "HTML5 in Action". Manning Publications, 2014. Pag.200.
- [Rob Hawkes, 2011] Rob Hawkes. "Foundation HTML5 Canvas". Apress, 2011. Pag.--
- [Robin Nixon, 2015]. Robin Nixon. "JavaScript: 20 Lessons to Successful Web Development". McGraw-Hill, 2015. Pag.--
- [shenoy et al, 2013] Aravind shenoy e Gianluca Guarini. "HTML5 and CSS3 Transitions, Transformations, and Animations", Pack Publishing, 2ª Edição, 2013 , pág.23
- shenoy et al (b), 2013] Aravind shenoy e Gianluca Guarini. "HTML5 and CSS3 Transitions, Transformations, and Animations", Pack Publishing, 2ª Edição, 2013 , pág.29
- [SnapSvg, 2015] SnapSvg. "About". <http://snapsvg.io/about/> [Último acesso: Setembro de 2015]
- [Steve Jobs, 2010] Steve Jobs. "Thoughts on Flash". <http://www.apple.com/hotnews/thoughts-on-flash/> [último acesso: Setembro de 2015]
- [Steve Patterson, 2015] Steve Patterson. "Managing Panels in Photoshop CS6", [ultima acesso: Agosto 2015]
- [Tiffany B. et al, 2014] Tiffany B. Brown, Kerry Butters, Sandeep Panda. " Jump Start - HTML5". SitePoint, 2014. Pag.-
- [Tiffany B. et al (b), 2014] Tiffany B. Brown, Kerry Butters, Sandeep Panda. " Jump Start - HTML5". SitePoint, 2014. Pag.176.
- [Tiffany B. et al (c), 2014] Tiffany B. Brown, Kerry Butters, Sandeep Panda. " Jump Start - HTML5". SitePoint, 2014. Pag.177.
- [Tom Green, 2015] Tom Green, Silvia Pfeiffer. "Beginning HTML5 Media". Apress, 2ª edição, 2015. Pag.
- [Victor M., 2012] Victor M. "Adobe não suportará Flash em Android 4.1". <http://pplware.sapo.pt/informacao/adobe-nao-suportara-flash-no-android-4-1/> [último acesso: Setembro de 2015]
- [TypeKit, 2014]. TypeKit, *Adobe Typekit Plans*, <https://typekit.com/plans> [ultimo 13/08/2015]
- [Yassine Bouhiel, 2010] Yassine Bouhiel. "A nostalgic Rummage Throught the History of Flash". <http://code.tutsplus.com/articles/a-nostalgic-rummage-through-the-history-of-flash--active-6733> [Último acesso: setembro de 2015]
- [W3, 2015] W3. "HTML5 is a Recommendation". <http://www.w3.org/blog/news/archives/4167> [último acesso: Setembro de 2015]
- [W3 (b), 2015], W3. "Canvas Text – best practice". <http://www.w3.org/TR/2dcontext/#text-styles> [Último acesso: Setembro de 2015]
- [W3C, 2015] W3C. "DOM". <http://www.w3.org/TR/WD-DOM-971209/> [último acesso: Setembro 2015]
- [W3C (b), 2015] W3C. "HTML5 - Developer view - History". <http://www.w3.org/TR/html5/introduction.html#history-0> [último acesso: Setembro de 2015]
- [W3C (c), 2015] W3C. "The CSS Saga". <http://www.w3.org/Style/LieBos2e/history/> [Último Acesso: Setembro de 2015].
- [W3Shools,2015] W3Shools. "HTML Introduction". http://www.w3schools.com/html/html_intro.asp [último acesso: Setembro de 2015]

- [W3Schools (b), 2015] W3Schools. "SVG Tutorial". <http://www.w3schools.com/svg/default.asp> [último acesso: Setembro de 2015]
- W3Schools (c), 2015] W3Schools. "SVG in HTML". http://www.w3schools.com/svg/svg_inhtml.asp [último acesso: Setembro de 2015]
- [W3Schools (d), 2015] W3Schools. "CSS Introduction". [Http://www.w3schools.com/css/css_intro.asp](http://www.w3schools.com/css/css_intro.asp) [último acesso: Setembro de 2015]
- [WHATWG, 2015] WHATWG. "HTML - History". <https://html.spec.whatwg.org/multipage/introduction.html#is-this-html5?> [último acesso: Setembro de 2015]
- [WHATWG (b), 2015] WHATWG. "FAQ, "What is the WHATWG?". <https://wiki.whatwg.org/wiki/FAQ> [último acesso: Setembro de 2015]
- [Wikibooks, 2015] Wikibooks, "Introduction to Computer Information System /n Multimedia". https://en.wikibooks.org/wiki/Introduction_to_Computer_Information_Systems/Multimedia [Último Acesso: Setembro 2015]
- [Rod Lawton, 2014] Rod Lawton. *Adobe Photoshop CC 2014 Review*, [Último acesso: Agosto 2015]
- [Yassine Bouhlel, 2010] Yassine Bouhlel. "A Nostalgic Rummage Through the History of Flash". <http://code.tutsplus.com/articles/a-nostalgic-rummage-through-the-history-of-flash--active-6733> [último acesso: Setembro 2015]

Anexos 1

(Inquérito para perfil de utilizador)

Recolha de dados para Tese de Mestrado

Obrigada pela sua disponibilidade para participar!

Este estudo no âmbito da Tese de Mestrado Engenharia Informática - Sistema Gráficos e Multimédia.

Pretende-se conhecer o tipo de utilizador no contexto da ferramentas multimédia, tecnologia html5, bem como que tipo de sistemas operativos e dispositivos utilizados hoje em dia.

As suas respostas serão estritamente confidenciais e muito importantes para um adequado enquadramento da Tese .

Agradeço desde já o preenchimento deste questionário até dia 19 de Junho.

Muito Obrigada!

Agostinho Monteiro

[Continuar »](#)

 12% concluído

Com tecnologia  Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 72 - Anexo 1: Formulário 1 (Pág. 1)

Recolha de dados para Tese de Mestrado

*Obrigatório

Perfil de Utilizador

Idade? *

- [21 - 25]
- [26 - 30]
- [31 - 40]
- [+40]


Sexo? *

- M
- F

« Anterior

Continuar »



Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 73 - Anexo 1: Formulário 1 (Pág. 2)

Recolha de dados para Tese de Mestrado

*Obrigatório

Software & SO - Desktop

Qual o Sistema Operativo Desktop que Utiliza? *

- Microsoft Windows
- Apple
- Outro Unix Based

Já trabalhou, a nível académico ou Profissional, Editores de imagem? *

- Sim
- Não

« Anterior

Continuar »

 37% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 74 - Anexo 1: Formulário 1 (Pág. 3)

Recolha de dados para Tese de Mestrado

*Obrigatório

Software & SO - Desktop (Cont.)


Quais dos seguintes editores já trabalhou ou testou? *

- Adobe Photoshop
- Gimp
- Microsoft Windows - Paint
- Paint.Net
- Corel PhotoPaint
- Outra:

« Anterior

Continuar »

 50% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 75 - Anexo 1: Formulário 1 (Pág. 4)

Recolha de dados para Tese de Mestrado

*Obrigatório

Software & SO - Online

Já trabalhou ou testou algum editor de imagem Online? *

Sim


Não

« Anterior

Continuar »



62% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 76 - Anexo 1: Formulário 1 (Pág. 5)

Recolha de dados para Tese de Mestrado

*Obrigatório

Software & SO - Online (Cont.)

Quais dos seguintes editores Online já trabalhou ou testou? *

- SumoPaint
- AutoDesk Pixlr
- Picozu
- Outra:


Qual o Browser que utiliza mais neste momento? *

- IE
- Mozilla Firefox
- Google Chrome
- Safari
- Opera
- Outra:

« Anterior

Continuar »

 75% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 77 - Anexo 1: Formulário 1 (Pág. 6)

Recolha de dados para Tese de Mestrado

*Obrigatório

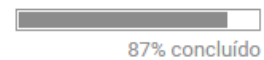
Software & SO - Smartphone & Tablet

Já testou algum editor de imagem num smartphone / tablet? *


- Sim
- Não

« Anterior

Continuar »



87% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 78 - Anexo 1: Formulário 1 (Pág. 7)

Recolha de dados para Tese de Mestrado

*Obrigatório

Software & SO - Smartphone & Tablet (Cont.)

Quais dos seguintes editores tem ou já teve no seu smartphone ou Tablet?

Fotor

Adobe Pixlr

Outra:

Qual o Sistema Operativo do seu smartphone or Tablet? *

Android

iOS

Windows Phone / Mobile

Outra:

« Anterior

Enviar

100%: terminou.

Nunca envie palavras-passe através dos Formulários do Google.

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 79 - Anexo 1: Formulário 1 (Pág. 8)

Anexos 2

(Inquérito para Avaliação do Protótipo)

Recolha de dados sobre Teste do Protótipo

Obrigada pela sua disponibilidade para participar!

Este estudo no âmbito da Tese de Mestrado Engenharia Informática - Sistema Gráficos e Multimédia.

Pretende-se recolher informação sobre teste efectuado no protótipo para Tese de mestrado, bem como futuramente poder melhorar e corrigir os possíveis erros que possam surgir.

As suas respostas serão estritamente confidenciais, muito importantes e de suporte para o âmbito anteriormente referido.

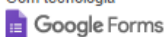
Agradeço desde já o preenchimento deste questionário até dia 19 de Junho.

Muito Obrigada!

Agostinho Monteiro

[Continuar »](#)

12% concluído

Com tecnologia  Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 80 - Anexo 2: Formulário 2 (Pág. 1)

Recolha de dados sobre Teste do Protótipo

*Obrigatório

Formulário Aplicação - Geral

Idade? *

- [21 - 25]
- [26 -30]
- [31-40]
- [+40]

Sexo? *

- M
- F


Visualizou o video de demonstração da aplicação? *

- Sim
- Não

« Anterior

Continuar »

 25% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 81 - Anexo 2: Formulário 2 (Pág. 2)

Recolha de dados sobre Teste do Protótipo


*Obrigatório

Formulário Aplicação - Desktop

Em que sistema Operativo testou a aplicação no Desktop? *

- Windows
- Apple
- Outro Unix Based
- Não testei em Desktop

[« Anterior](#) [Continuar »](#)

 37% concluído

Com tecnologia  Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 82 - Anexo 2: Formulário 2 (Pág. 3)

Recolha de dados sobre Teste do Protótipo

*Obrigatório

Desktop - Usabilidade

Continuação formulário Aplicação - Desktop

Quais dos seguintes browsers (Navegadores) testou a aplicação no Desktop? *

- Google Chrome
- Mozilla Firefox
- Opera
- Safari
- IE
- Outra:

Quanto fácil ou difícil é de utilizar a aplicação no Desktop? *

1 2 3 4 5

Muito Difícil Extremamente Fácil

Quanto é visualmente atraente é a interface Gráfica no Desktop? *


1 2 3 4 5

Nada Muito

« Anterior

Continuar »

 50% concluído

Com tecnologia


Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 83 - Anexo 2: Formulário 2 (Pág. 4)

Recolha de dados sobre Teste do Protótipo

*Obrigatório

Desktop - Funcionalidades

Continuação formulário Aplicação - Desktop

Quanto a aplicação tem funcionalidades capazes de colmatar as suas necessidades? *

1 2 3 4 5

Nada capazes Extremamente capazes

Conseguiu executar todas as tarefas desejadas? *

1 2 3 4 5

nenhuma todas

Qual das seguintes funcionalidades gostaria ver implementada? *

- Crop
- Texto
- Imagens por URL
- Drag & Drop
- Ferramenta de molduras
- Combinar múltiplas fotos
- Undo / desfazer
- Outra:

Voltaria a usar a aplicação em Desktop? *

- Sim
- Não

Sugestões ou erros que possam ter ocorrido

Opcional

« Anterior

Continuar »

62% concluído

Figura 84 - Anexo 2: Formulário 2 (Pág. 5)

Recolha de dados sobre Teste do Protótipo

*Obrigatório

Formulário Aplicação - Smartphone / Tablet


Em que Sistema Operativo do Smartphone / Tablet testou aplicação? *

- Windows
- Apple
- Android
- Firefox OS
- Ubuntu Phone
- Não testei em Smartphone / Tablet

« Anterior

Continuar »

 75% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 85 - Anexo 2: Formulário 2 (Pág. 6)

Recolha de dados sobre Teste do Protótipo

*Obrigatório

Smartphone / Tablet - Usabilidade

Em que browser (navegador) testou no seu smarphone / Tablet? *

- Default Android browser
- Google chrome
- Mozilla Firefox
- Opera
- Opera Mini
- Safari
- IE
- Outra:

Quanto fácil ou difícil é de utilizar a aplicação num smartphone / tablet? *

1 2 3 4 5

Muito Difícil Extremamente Fácil

O quanto é visualmente atraente a interface Gráfica smartphone / tablet? *


1 2 3 4 5

Nada Muito

« Anterior

Continuar »

 87% concluído

Com tecnologia
 Google Forms

Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 86 - Anexo 2: Formulário 2 (Pág. 7)

Smartphone / Tablet - Funcionalidades

Continuação do formulário Aplicação - Smartphone / Tablet

O quanto a aplicação tem funcionalidades capazes de colmatar as suas necessidades? *

1 2 3 4 5

Nada capazes Extremamente capazes

Conseguí executar todas as tarefas desejadas? *

1 2 3 4 5

nenhuma todas

Qual das seguintes funcionalidades gostaria ver implementada? *

- Crop
- Texto
- imagem por URL
- Drag & Drop
- Ferramenta de Molduras
- Combinar Múltiplas fotos
- Outra:

Voltaria a usar a aplicação num smartphone / tablet? *

- Sim
- Não

Sugestões ou erros que possam ter ocorridos.


Opcional

« Anterior

Enviar

100%: terminou.

Nunca envie palavras-passe através dos Formulários do Google.

Com tecnologia


Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Figura 87 - Anexo 2: Formulário 2 (Pág. 8)