



Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Dário Miguel dos Santos Mendes

Estágio na Compta – Business Solutions

Relatório de Estágio

Orientado por:

Professor Doutor Pedro Correia, IPT – Instituto Politécnico de Tomar
Daniel Silva – Compta S.A.

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

Dedicado aos meus pais, que sempre me apoiaram incondicionalmente

RESUMO

O presente documento descreve o trabalho realizado no segundo ano letivo do Mestrado em Produção de Conteúdos Digitais. Relatando um estágio decorrido na empresa tecnológica Compta S.A.

Neste relatório é feita uma apresentação da empresa, bem como os seus objetivos como tal. De seguida, efetua-se um enquadramento do estágio no âmbito contextual dos conteúdos digitais, descrevendo as tecnologias utilizadas, necessárias à criação de aplicações para dispositivos móveis. São ainda apresentados os projetos desenvolvidos ao longo do estágio na empresa, fundamentando os aspetos que mais se enquadram com o plano de estudos do Mestrado.

Palavras-chave: AngularJS, Aplicação Móvel, Conteúdos Digitais, Cordova, CSS, HTML, JavaScript.

ABSTRACT

This document describes the work done in the second academic year of the Master in Digital Content Production. Reporting an internship at Compta S.A.

This report gives a presentation of the company as well as its main goals as such. Next, a contextual framework of the digital content is described, describing the technologies used, necessary for the creation of applications for mobile devices. Also a presentation of the projects developed during the internship in the company, based on the aspects that best fit the Master's study plan.

Keywords: AngularJS, Mobile Application, Digital Content, Cordova, CSS, HTML, JavaScript.

AGRADECIMENTOS

Em primeiro lugar, deixo um especial agradecimento à empresa Compta. Tanto por me ter dada esta oportunidade de estágio curricular, como pela hospitalidade e as ajudas fornecidas na sua duração.

De seguida, pretendo agradecer aos membros da Compta, Daniel Silva e Nuno Miquelina, pela paciência e pelo enorme apoio dado nestes nove meses de estágio. Agradecendo ainda ao Professor Doutor Pedro Correia, pela ajuda e orientação dada durante a concretização do deste relatório.

Ao Oleksandr Novak pela cooperação e entreajuda em determinados aspetos dos projetos desenvolvidos. E ainda aos restantes colaboradores da Compta de Abrantes.

Quero deixar uma palavra de consideração a todos os docentes do mestrado, que sempre me apoiaram e prepararam de maneira a conseguir superar os obstáculos encontrados ao longo deste estágio.

Também desejo agradecer à Professora Doutora Sandra Jardim por ter originado esta oportunidade de estágio.

A estes, e a todos os que, de alguma forma, também contribuíram para a concretização desta etapa da minha vida, um forte e sentido obrigado.

ÍNDICE

RESUMO	vii
ABSTRACT	ix
AGRADECIMENTOS	xi
Índice de Figuras	XVII
Índice de Tabelas	XIX
Índice de Código.....	XXI
Lista de Abreviatura e Siglas.....	XXIII
1 Introdução	1
1.1 Objetivo pretendido	1
1.2 Estrutura do documento	1
2 Compta	3
2.1 A Empresa.....	3
2.2 Local de trabalho	5
3 Enquadramento	7
3.1 Tecnologias Utilizadas.....	7
3.1.1 AngularJS	7
3.1.2 Cordova	11
3.1.3 CSS3.....	12
3.1.4 Drupal.....	14

3.1.5	HTML5	14
3.1.6	Ionic	16
3.1.7	JavaScript.....	19
3.1.8	jQuery	21
3.1.9	JSON.....	22
3.1.10	PHP	23
3.1.11	SQL	23
3.1.12	XML.....	25
3.2	Ferramentas Utilizadas	26
3.2.1	Browsers	26
3.2.2	Editores de texto	27
3.2.3	Outros.....	28
3.2.4	Dispositivos.....	28
4	Metodologia	31
4.1	Projeto – Sou Cidadão	31
4.1.1	Serviço – Ocorrências	33
4.1.2	Serviço - Orçamento Participativo.....	35
4.1.3	Serviço – Recolha de Monos e Verdes	37
4.1.4	Notificações	38
4.2	Projeto – PaxVoice	42
4.2.1	Serviço – Registo e Login por Email.....	44

4.2.2	Serviço – Definições de Utilizador	54
4.3	Dificuldades	56
5	Conclusões.....	57
6	Bibliografia.....	59

Índice de Figuras

<i>Figura 1 - Logotipo do Grupo Compta.....</i>	<i>3</i>
<i>Figura 2 - Organograma do Grupo Compta</i>	<i>4</i>
<i>Figura 3 – Instalações TAGUSVALLEY.....</i>	<i>5</i>
<i>Figura 4 - Logotipo AngularJS.....</i>	<i>10</i>
<i>Figura 5 - Arquitetura geral de uma aplicação Cordova.....</i>	<i>11</i>
<i>Figura 6 - Logotipo Cordova.....</i>	<i>12</i>
<i>Figura 7 - Estrutura e evolução tecnológica CSS</i>	<i>13</i>
<i>Figura 8 - Logotipo CSS3</i>	<i>13</i>
<i>Figura 9 - Logotipo Drupal</i>	<i>14</i>
<i>Figura 10 - Logotipo HTML5</i>	<i>15</i>
<i>Figura 11 - Hello World em Ionic.</i>	<i>19</i>
<i>Figura 12 - Logotipo Ionic</i>	<i>19</i>
<i>Figura 13 - Logotipo JavaScript.....</i>	<i>20</i>
<i>Figura 14 - Logotipo jQuery.....</i>	<i>21</i>
<i>Figura 15 - Logotipo JSON</i>	<i>22</i>
<i>Figura 16 - Logotipo PHP</i>	<i>23</i>
<i>Figura 17 - Logotipo SQL.....</i>	<i>25</i>
<i>Figura 18 - Logotipo XML.....</i>	<i>26</i>
<i>Figura 19 - Logotipo Sou Cidadão</i>	<i>31</i>

<i>Figura 20 - Arquitetura do Projeto Sou Cidadão</i>	<i>32</i>
<i>Figura 21 - Ecrã inicial e menu lateral.....</i>	<i>33</i>
<i>Figura 22 - Imagens marcadores de agrupamento.</i>	<i>34</i>
<i>Figura 23 - Exemplo agrupamento de marcadores.....</i>	<i>34</i>
<i>Figura 24 - Exemplos de elementos desativados de acordo com a fase OP.</i>	<i>36</i>
<i>Figura 25 - Exemplo de boletim de voto para um máximo de três votos.....</i>	<i>37</i>
<i>Figura 26 - Ecrãs de recolha de monos e verdes.</i>	<i>38</i>
<i>Figura 27 – Configuração de variáveis do método MQsendPush no Drupal.....</i>	<i>40</i>
<i>Figura 28 - Exemplo de notificação.</i>	<i>42</i>
<i>Figura 29 - Arquitetura do projeto PaxVoice</i>	<i>43</i>
<i>Figura 30 - Logotipo PaxVoice</i>	<i>44</i>
<i>Figura 31 - Ecrã inicial PaxVoice.....</i>	<i>45</i>
<i>Figura 32 - Ecra de autenticação.....</i>	<i>46</i>
<i>Figura 33. - Erros de validação no processo de autenticação</i>	<i>47</i>
<i>Figura 34 - Erros de validação no processo de registo.</i>	<i>48</i>
<i>Figura 35 - Processo de introdução de imagem no registo.</i>	<i>49</i>
<i>Figura 36 - Ecrã de registo efetuado com sucesso.....</i>	<i>51</i>
<i>Figura 37 - Ecrã de pedido de reposição de password.....</i>	<i>52</i>
<i>Figura 38 - Ecrã de confirmação de Email.....</i>	<i>53</i>
<i>Figura 39 - Ecrã Menu após login.</i>	<i>53</i>
<i>Figura 40 - Ecrã Definições de utilizador.....</i>	<i>55</i>

Índice de Tabelas

Tabela 1 - Diretivas AngularJS.....	9
Tabela 2 - Exemplo de tabela criada.....	25

Índice de Código

Código 1 - HTML de exemplo Ionic.	17
Código 2 – JavaScript de exemplo Ionic.	18
Código 3 - Exemplo JSON.	22
Código 4 – Criação da estrutura de Tabela SQL.	24
Código 5 - Inserção de valores na tabela.	24
Código 6 - Seleção a todos os valores da tabela.	24
Código 7 – Exemplo XML	26
Código 8 - CSS referente à classe disabled.	36
Código 9 - Função push com base no tipo de plataforma.	41

Lista de Abreviatura e Siglas

API – Application Programming Interface

CMS – Content Managing System

CSS – Cascading Style Sheets

DOM – Document Object Model

GIF – Graphics Interchange Format

HTML – Hyper Text Markup Language

iOS – iPhone Operating System

JPEG - Joint Photographic Experts Group

JSON - JavaScript Object Notation

PHP – Originalmente Personal Home Page, atualmente Hypertext Preprocessor

PNG – Portable Network Graphics

SQL – Structured Query Language

UI – User interface

WP8 – Windows Phone 8

WS – WebService

XML - eXtensible Markup Language

1 Introdução

O presente documento surge na sequência dos estudos do Mestrado em Produção de Conteúdos Digitais no âmbito do estágio, ministrado pelo Instituto Politécnico de Tomar. Contando com a orientação do Doutor Pedro Correia por parte do IPT, bem como de Daniel Silva e Nuno Miquelina, membros da Compta S.A..

Este estágio curricular teve uma duração de nove meses. Pelo que terá decorrido em Abrantes, nomeadamente no Tecnopolo - TAGUSVALLEY. Onde se encontram as instalações pertencentes ao Grupo Compta.

1.1 Objetivo pretendido

O estágio curricular tem como objetivo a aplicação dos conhecimentos multisiaciplinares adquiridos durante primeiro ano do Mestrado em Produção de Conteúdos Digitais. Sendo que cabe ao estagiário desempenhar a função de programador para as tarefas e desenvolvimentos pretendidos em determinados projetos da empresa. Trabalhando em equipa num ambiente profissional e responsável, respeitando horários de trabalho e prazos de entregas.

1.2 Estrutura do documento

A estrutura deste relatório de estágio subdivide-se nos seguintes capítulos:

- **Introdução** – Apresenta uma breve descrição do âmbito do estágio e dos projetos nele desenvolvidos. Sumarizando a as suas implementações e principais tecnologias utilizadas.
- **Compta** – Breve apresentação da empresa, especificando as suas origens, objetivos, certificados e parcerias.
- **Enquadramento** – Descrição de cada uma das tecnologias e ferramentas utilizadas na conceção dos projetos.
- **Metodologia** – Descrição mais detalhada do desenvolvimento do projetos Sou Cidadão e PaxVoice. São apresentadas as soluções encontradas na resolução das tarefas e expostas as maiores dificuldades e contratempos.

- **Conclusões** – Apreciação final do estágio realizado e respectivos objetivos concretizados.

2 Compta

Neste capítulo é feita uma descrição da empresa onde terá decorrido o estágio. Analisando a sua estrutura organizacional, posição nos mercados nacionais e internacionais, bem como os seus objetivos como empresa.



Figura 1 - Logotipo do Grupo Compta

2.1 A Empresa

A Compta – Equipamentos e Serviços de Informática S.A. terá sido fundada em 1972 em Lisboa. Sendo um dos mais antigos Grupos Tecnológicos Nacionais, contando com 45 anos de existência[1]. Inicializou as suas operações como um Centro de Processamento de Dados, formado especialmente com a finalidade de prestação de serviços aos seus acionistas originais. Desde então, acompanhando o rápido desenvolvimento da indústria das tecnologias de informação, a Compta evoluiu ao longo dos anos. Principalmente no que toca ao desenvolvimento de soluções nas áreas de telecomunicações e sistemas de processamento, bem como à comercialização de produtos. É uma empresa especializada em adaptar as melhores soluções tecnológicas de fabricantes mundiais, aos mercados onde atua, introduzindo uma forte componente local de engenharia e gestão de projetos. Apresenta uma vasta gama de produtos, soluções e serviços, que impulsionam o desempenho e atividade das empresas em diferentes mercados: Ambiente e Resíduos; Administração Pública; Agricultura; Cidades e IoT (Internet of Things); Energia e Utilities; Industria & Serviços; Mar e Crescimento Azul; Saúde; Serviços Financeiros; Telecomunicações; Transportes[1];

Atualmente, a Compta S.A conta com várias parcerias em diferentes áreas tecnológicas. Em especial com a gigante americana IBM. Possuindo o mais elevado grau de parceria,

para o desenvolvimento de soluções no âmbito das Smart Cities, IoT, Analytics, Commerce e serviços Cloud. A Compta S.A. é composta por mais de 240 colaboradores, onde mais de 70% são técnicos especializados nas áreas onde atuam.

Presentemente, a empresa divide-se em 5 ramificações:

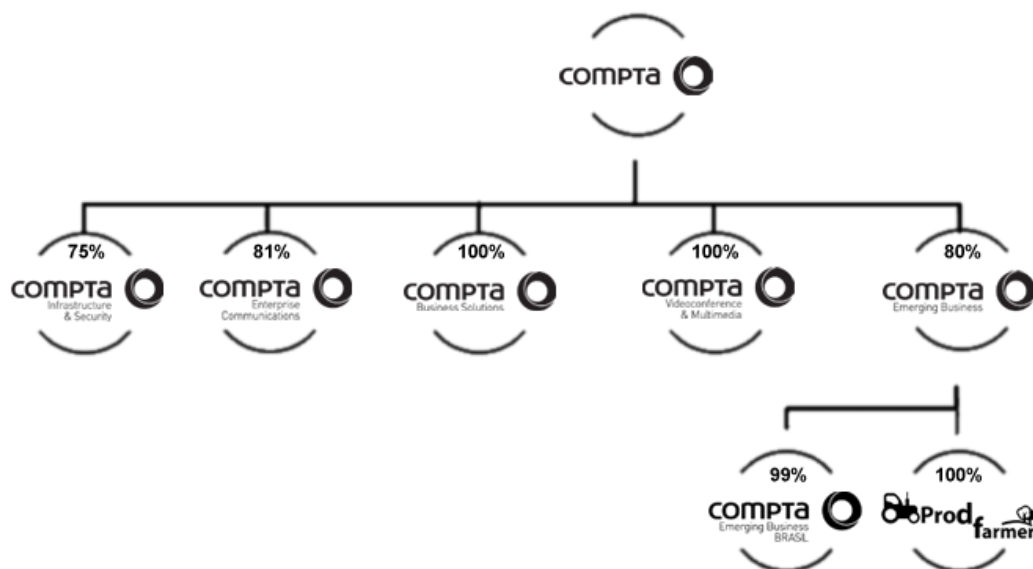


Figura 2 - Organograma do Grupo Compta

A Compta S.A. acumula mais de 400 certificações individuais e detém o mais elevado nível de certificação nos seus principais parceiros estratégicos. Possui a certificação em Sistemas de Gestão de Serviços de Tecnologias de Informação (ITSMS – Information Technology Service Management System) de acordo com o referencial ISO/IEC 20000-1:2011 para o âmbito “Prestação de serviços no mercado português, no domínio da parametrização aplicacional e suporte de plataformas destinadas a soluções de Gestão de Processos, a partir de Portugal”[1].

É ainda um Grupo certificado segundo as normas – NP 4457 – Sistema de Gestão da Investigação, Desenvolvimento e Inovação e ISO/IEC 27001:2013 (Sistema de Gestão da Segurança da Informação) para além da NP EN ISO 14000 Certificação Ambiental e NP EN ISO 9001, modelo de garantia da qualidade na conceção/desenvolvimento, produção, instalação e assistência pós-venda, no âmbito do fornecimento de produtos e soluções integradas de Telecomunicações e Tecnologias de Informação[1].

2.2 Local de trabalho

Apesar da sede da Compta S.A. ser localizada em Lisboa. O presente estágio decorreu nas instalações da empresa em Alferrarede, concelho de Abrantes. Nomeadamente no TAGUSVALLEY - Tecnopolo do Vale do Tejo. Este foi uma aposta da Câmara Municipal de Abrantes, e o apoio do Instituto Politécnico de Tomar, com o objetivo de promover o empreendedorismo e a competitividade da região[2]. Este Tecnopolo está envolvido nos sectores das Tecnologias da Informação e Comunicação, Energia, Metalomecânica e Agroalimentar. Contando com mais de 32 empresas nas suas instalações desde o ano de



Figura 3 – Instalações TAGUSVALLEY

2015.

As instalações da Compta neste parque tecnológico providenciam um local de trabalho agradável, com boas condições e equipamentos, favoráveis ao desenvolvimento e produção de diferentes projetos. Neste espaço encontram-se colaboradores do grupo Compta – Business Solutions (CBS) e Compta – Emerging Business (CEB). Ambos possuindo equipas compostas por pessoas com excelente espírito de equipa e entreaduza.

3 Enquadramento

A informação digital disponibilizada na World Wide Web tem aumentado consideravelmente ao longo dos anos. Isto devido à quantidade de dados gerados ser cada vez maior[5]. Aliado a este aumento, está uma evolução constante nos avanços tecnológicos, nomeadamente por parte dos dispositivos móveis. De maneira a fazer face a esta evolução, é necessário que as plataformas e interfaces de acesso à informação se adaptem constantemente. Para que desta forma, tanto os utilizadores, como os desenvolvedores possam facilmente interagir com essa dada informação.

Cabe ao produtor de conteúdos digitais, o papel de garantir que as informações sejam devidamente estruturadas e organizadas. Possibilitando melhoramentos no que toca à acessibilidade e métodos de interação entre o utilizador e os conteúdos.

Como forma de enquadrar tecnologicamente o trabalho de estágio desenvolvido, os próximos subcapítulos descrevem toda a base de trabalho utilizada.

3.1 Tecnologias Utilizadas

De seguida, são apresentadas e descritas as várias tecnologias e ferramentas utilizadas nos projetos ao longo do estágio. Nomeadamente, linguagens de programação, *frameworks* e bibliotecas.

3.1.1 AngularJS

O AngularJS é uma *framework* para aplicações web baseada em JavaScript, desenvolvida e mantida pela Google. Possui bastantes componentes de JavaScript que complementam a *framework* Apache Cordova, dedicada ao desenvolvimento de aplicações móveis multiplataforma, apresentada no próximo subcapítulo. O AngularJS visa simplificar tanto o desenvolvimento, como os testes dessas aplicações, fornece uma estrutura com base nas arquiteturas model-view-controller (MVC) e model-view-viewmodel (MVVM) do lado do cliente[3].

Esta tecnologia começa por ler o ficheiro HTML, onde se encontram *tags* especiais, que o Angular interpreta como diretivas. Desta forma é criada a ligação entre a apresentação (inputs e outputs) da página e o seu modelo.

O AngularJS utiliza um termo “scope” que serve como elo de ligação entre o HTML (View) e o JavaScript (Controller). O scope representa um objeto (que segue as regras normais de um objeto JavaScript) com propriedades e métodos, podendo ser acedido e manipulado tanto por parte da View, como pelo Controller. Seguindo a arquitetura MVC, o scope representa assim o Model.

O *Bootstrap*, ou seja, o processo de inicialização do AngularJS *bootstrapper* ocorre em três fases assim que o DOM finaliza o seu carregamento:

1. Criação de um novo injetor
2. Serviço de compilação - onde é efetuada a leitura do DOM e localizadas todas as diretivas relativas ao AngularJS.
3. Link – fase de ligação, onde são anexadas todas as diretivas ao scope.

As diretivas do AngularJS permitem especificar *tags* HTML personalizadas e reutilizáveis. Estas personalizam o comportamento dos seus elementos e atributos, que definem a sincronização entre o Model e a View, bem como o comportamento e a apresentação dos seus componentes.

De seguida, apresentam-se as principais diretivas AngularJS:

Diretiva	Descrição
ng-app	Declara um elemento como o elemento raiz da aplicação, gerando a mudança do comportamento padrão do elemento.
ng-bind	Altera o texto de um elemento HTML automaticamente, de acordo com o seu resultado, respeitando regras estabelecidas.
ng-model	Similar ao ng-bind, no entanto permite ligação direta bidirecional (two-way data binding) entre a view e o scope da aplicação.

ng-class	Permite atributos de classe serem atribuídos dinamicamente
ng-click	Permite instanciar o evento de <i>click</i> , semelhante ao <i>onclick</i> .
ng-controller	Especifica um controller JavaScript para um determinado HTML em questão.
ng-repeat	Instancia um elemento por item de um array.
ng-show ng-hide	Mostra ou esconde um elemento HTML de acordo com o resultado de uma expressão booleana.
ng-switch	Instancia um template numa lista de escolhas, dependendo do valor obtido pela expressão.
ng-view	A diretiva base para manipulação de rotas, resolvendo um JSON antes de efetuar o <i>rendering</i> dos modelos acionados por controladores específicos.
ng-if	Declaração básica de um 'if' que permite mostrar um elemento caso a condição seja verdadeira.

Tabela 1 - Diretivas AngularJS.

O AngularJS possui uma ligação bidirecional de dados (Two-way data binding), que reduz a quantidade de código escrito, para mostrar os dados processados pelo servidor. Existem *templates* que são carregados em HTML puro, de acordo com os dados contidos num scope definido num Model. A variável *\$scope* do Angular deteta alterações no Model e altera o HTML na View recorrendo a um Controller. Da mesma forma, qualquer alteração efetuada na View é refletida no Model. Estas características excluem a necessidade de manipular o DOM, facilitando o *bootstrapping* melhorando assim o processo de desenvolvimento de aplicações web e móveis.

Nos projetos desenvolvidos foi utilizada uma tecnologia baseada em AngularJS, denominada Angular Material. É considerada uma *framework* de componente da interface de utilizador(UI), bem como uma referência da especificação do Material Design criado e mantido pela Google.

O Angular Material possui uma série de componentes UI reutilizáveis por toda a aplicação, destacando-se[4]:

- **Autocomplete** (<md-autocomplete>) – Incorpora uma caixa de texto que fornece uma lista de possíveis resultados, provenientes de uma base de dados, com formato de dropdown.
- **Datepicker** (<md-datepicker>) – Ferramenta que permite ao utilizador seleccionar uma data específica a partir de um calendário gerado automaticamente, possuindo vários tipos de estilos.
- **Dialog** (<md-dialog>) – Representa um alerta que poderá ser apenas informativo, ou solicitar informações ao utilizador, tais como confirmações e preenchimento de caixas de texto.
- **Select** (<md-select>) – Estende as funcionalidades de uma tag select HTML. permite seleccionar múltiplas opções através de checkboxes incorporadas no dropdown, bem como popular o próprio select através de pedidos assíncronos, mostrando um indicador de carregamento da informação.

Para além das ferramentas fornecidas pelo Angular Material, uma das suas principais vantagens é ser altamente compatível com aplicações móveis. Pois as suas componentes UI escalam dinamicamente de acordo com o tamanho de ecrã disponível.



Figura 4 - Logotipo AngularJS

3.1.2 Cordova

O Apache Cordova, originalmente chamado PhoneGap, é uma framework open source de desenvolvimento de aplicações móveis. Permite a utilização de HTML5, CSS3 e JavaScript, tornando-se numa tecnologia bastante útil no desenvolvimento nas aplicações multiplataforma. Podendo assim ser uma alternativa viável à utilização de API que são específicas e exclusivas para cada plataforma (Android, iOS, ou Windows Phone).

Esta framework efetua o *wrapping* do código. Ou seja, com base no código CSS, HTML e JavaScript fornecido é gerada a aplicação dependendo da plataforma do dispositivo. As aplicações que resultam deste processo são consideradas híbridas[7]. Não são aplicações móveis completamente nativas, pois o layout da aplicação não é gerado com base na framework UI nativa de cada plataforma, mas sim com Web Views[7]. Nem são totalmente baseadas em Web, pois podem ser geradas para distribuição e são aplicações que possuem acesso às API's nativas de cada dispositivo.

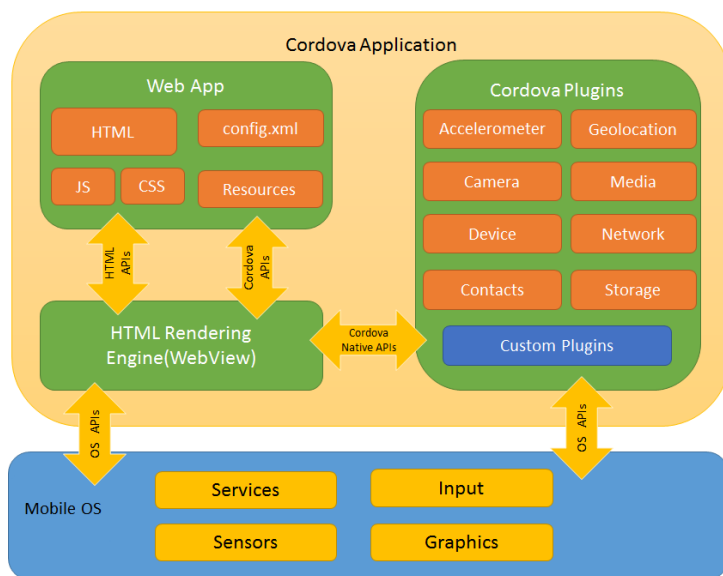


Figura 5 - Arquitetura geral de uma aplicação Cordova.

A WebView é responsável pela totalidade da interface do utilizador. Em certas plataformas, (Android iOS, Windows Phone) a WebView permite incorporar o uso de certos componentes de cada uma delas, tal como se tratasse de uma aplicação nativa[7].

A Web App é onde reside o código do projeto. A aplicação em si é implementada como uma página web. Por defeito existe um ficheiro chamado `index.html` que faz referência ao CSS, JavaScript e bibliotecas necessárias ao seu funcionamento. A componente Web App possui um ficheiro crucial denominado `config.xml`, fornecendo informação sobre a aplicação e possui parâmetros que definem o seu comportamento em aspetos mais básicos de cada plataforma[13]. Tais como orientação do ecrã ou definição de ecrãs iniciais.

A framework Apache Cordova possui uma série de plugins disponíveis no seu ecossistema. Estes plugins fornecem uma interface que gere a comunicação entre a própria framework e componentes nativos para cada plataforma, bem como as suas API's. Um projeto desenvolvido com base nesta framework, possui uma serie de plugins (Core Plugins) que possibilitam o acesso por parte da aplicação ao hardware do dispositivo, tais como bateria, camara, acelerómetro, bussola, microfone, etc[13]. Para além destes plugins, existem vários plugins desenvolvidos pela comunidade, que possuem funcionalidades específicas, nem sempre disponíveis para todas as plataformas. É possível pesquisar estes plugins no repositório npm do node.js, ou até mesmo proceder à criação um novo plugin de raiz.

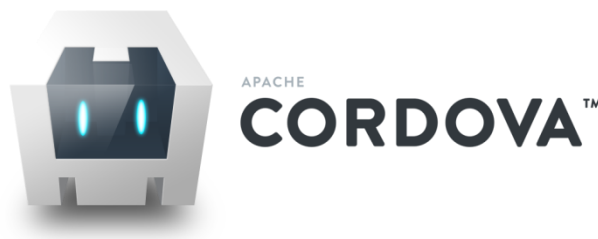


Figura 6 - Logotipo Cordova

3.1.3 CSS3

O CSS3 é a versão mais recente do CSS (*Cascading Style Sheets*). Esta tecnologia trata-se de uma linguagem do tipo *style sheet*, utilizada para descrever a apresentação de documentos escritos em linguagens do tipo *markup*, tais como HTML e XML[9].

A linguagem de CSS é projetada principalmente para permitir a separação e a apresentação de conteúdo. Nomeadamente o aspeto do layout, cores e fontes de uma página web. A importância desta separação do conteúdo é fornecer mais flexibilidade e controlo das interfaces de utilizador. Permite que várias páginas HTML compartilhem a mesma formatação e organização. O CSS é especificado num ficheiro do tipo “.css” e referenciado nos elementos(*tags*) HTML por intermedio de id's(#id) e classes(.class), de maneira a reduzir a complexidade e repetição do conteúdo estrutural.

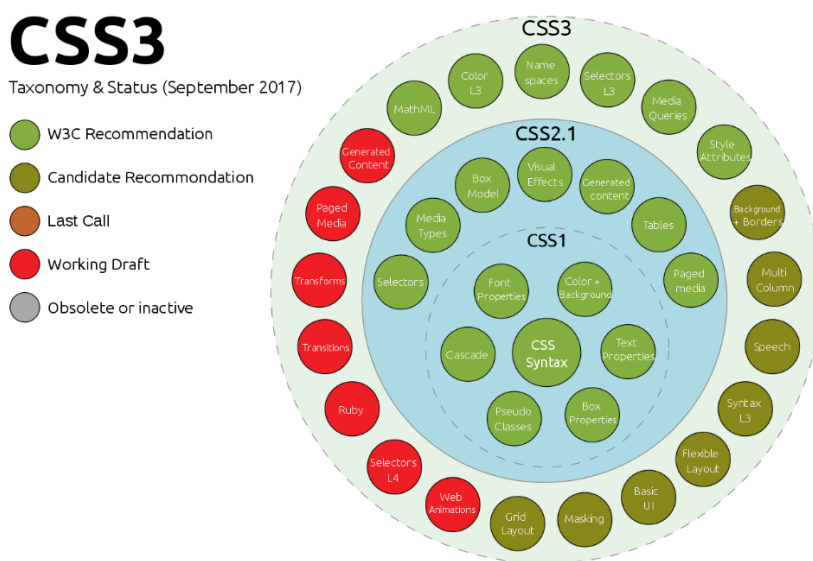


Figura 7 - Estrutura e evolução tecnológica CSS

No que toca ao CSS3, a principal novidade está presente na sua separação em módulos[9]. Continua a conter a antiga especificação de CSS, E foram adicionados novos módulos, tais como seletores (selecionar vários elementos onde se pretendem aplicar determinados



Figura 8 - Logotipo CSS3

estilos), animações, transformações 2D e 3D, efeitos de texto, etc.

3.1.4 Drupal

A tecnologia Drupal trata-se de uma framework modular e um sistema de gestão de conteúdo (Content Management System - CMS) desenvolvido em linguagem PHP. O Drupal permite criar e organizar conteúdos, manipular a estrutura e apresentação, automatizar tarefas administrativas e definir permissões e funções de utilizadores. Implementa uma série de API's robustas e apresenta uma estrutura modular que facilita o desenvolvimento de módulos extensivos.

O Drupal não necessita de sistema operativo devido ao facto de ser desenvolvido em PHP. No entanto, requer um servidor HTTP compatível com PHP, como o Apache, Nginx, Hiawatha ou Microsoft IIS[11]. É necessário também um servidor de base de dados, como por exemplo o MySQL ou Oracle.



Figura 9 - Logotipo Drupal

3.1.5 HTML5

O HTML (HyperText Markup Language) é uma linguagem do tipo marcação, responsável pela construção de conteúdos web. Os documentos do tipo HTML são interpretados pelos browsers, gerando páginas web multimédia. O HTML descreve a estrutura e a apresentação de páginas web[14].

O HTML5 é a versão mais atualizada da tecnologia HTML. As suas principais alterações e novas implementações são[14]:

- Sintaxe simplificada – permite desenvolvimento do código e do seu respetivo conteúdo de uma forma mais clara e precisa. Por exemplo, para declarar o DOCTYPE do documento, basta introduzir a seguinte sintaxe: `<!DOCTYPE html>`.
- Conteúdos multimédia – O HTML5 conta com suporte nativo de reprodução de ficheiros multimédia, tais como vídeo ou áudio. Nas versões anteriores seria necessário a utilização de plugins como o Flash ou o Silverlight.
- Armazenamento local – Permite o armazenamento em várias janelas, possuindo maior segurança e desempenho. Os dados locais podem ser mantidos mesmo após o browser ser encerrado. Resultando numa base de dados do lado do cliente, que permanece mesmo que o cliente elimine os cookies.
- Comunicação entre cliente e servidor - O HTML5 possui web sockets que possibilitam a comunicação full duplex entre cliente e servidor.
- Localização geográfica - Possui uma nota API GeoLocation permite que o cliente facilmente envie sua localização para uma determinada aplicação web. Por razões de privacidade e segurança, é requerido ao utilizador permissão para o envio de informações relativas à sua localização.
- Mecanismo de *threads* para JavaScript – Implementação do *Web Worker*. Sem a sua utilização, ao executar scripts numa página HTML, essa página não responde até que o script seja concluído. O *Web Worker* trata-se um JavaScript que é executado em segundo plano, independentemente de outros scripts, recorrendo às capacidades multi-core dos dispositivos atuais. Esta tecnologia melhora bastante o desempenho e performance no que toca à navegação em aplicações web.
- Gráficos e efeitos 2D/3D – Uma das maiores vantagens do HTML5, foi introduzido o suporte ao *canvas* que permite desenhar gráficos na página web. Estes gráficos podem são desenhados através do JavaScript. Foi também introduzido o suporte ao SVG (*Scalable Vector Graphics*) definido em formato XML, sendo que a principal vantagem deste formato de gráficos, é não perder qualidade caso seja aplicado



zoom ou redimensionamento. Ao nível de CSS3, foi também incluído o suporte a animações e transformações 2D/3D

3.1.6 Ionic

A tecnologia Ionic trata-se de um SDK baseado em AngularJS e desenvolvido sobre Apache Cordova. Tem como objetivo possibilitar o desenvolvimento de aplicações para dispositivos móveis. Tal como as aplicações desenvolvidas em Apache Cordova, o desenvolvimento Ionic também resulta em aplicações híbridas, recorrendo a CSS, HTML5 e JavaScript[7].

Vantagens da utilização do Ionic para o desenvolvimento de aplicações híbridas[8].

- Com o Ionic continua a ser possível utilizar qualquer comando ou plugin existente na framework Apache Cordova.
- O comando “*ionic serve*” é uma ferramenta extremamente útil ao nível de desenvolvimento, pois possibilita a edição do código e teste em tempo real. Assim que alguma alteração é efetuada, o protótipo da aplicação a ser executado no browser, é atualizado automaticamente.
- Sendo baseado em AngularJS, o Ionic permite recorrer à aceleração nativa do hardware do dispositivo, resultando em melhoramentos ao nível da performance da aplicação. Este processo é obtido através da otimização dos efeitos presentes nas animações, recorrendo às capacidades da unidade de processamento gráfico (GPU) do dispositivo.
- Inclui vários módulos e componentes UI (templates, ícones, botões, headers e footers).
- Dispõe de uma vasta documentação, clara e intuitiva.

No fundo, a tecnologia Ionic SDK atua como uma extensão da framework Apache Cordova. Fornecendo a uma aplicação híbrida, ou seja, que possui características bastante

similares a uma aplicação totalmente nativa. Tanto ao nível de design, como ao nível da performance.

Segue-se um exemplo básico para uma aplicação baseada em Ionic:

```
<!DOCTYPE html>
<html>
  <head>
    <!--Bibliotecas Ionic -->
    <link href="http://code.ionicframework.com/1.3.3/css/ionic.min.css" rel="stylesheet">
    <script src="http://code.ionicframework.com/1.3.3/js/ionic.bundle.js"></script>
  </head>
  <body ng-app="exemploIonicApp">
    <ion-nav-view></ion-nav-view>
    <script id="templates/main.html" type="text/ng-template">
      <ion-header-bar class="bar-stable">
        <h1 class="title">Olá Mundo!</h1>
      </ion-header-bar>
      <ion-content class="padding">
        <p>Bem vindo!</p>
        <br>
        <a href="#/page2">Ir para a proxima página?</a>
      </ion-content>
    </script>
    <script id="templates/page2.html" type="text/ng-template">
      <ion-header-bar class="bar-stable">
        <h1 class="title">Página 2</h1>
      </ion-header-bar>
      <ion-content class="padding">
        <p>Bem vindo à segunda página!</p>
        <br>
        <a href="#/main">Voltar?</a>
      </ion-content>
    </script>
  </body>
</html>
```

Código 1 - HTML de exemplo Ionic.

A estrutura de uma aplicação Ionic segue os princípios do HTML. Dentro do elemento `body` é inserido o elemento “`<ion-nav-view>`” onde são implementadas todas as *templates* com o seu respetivo conteúdo. O Ionic armazena um histórico destas *templates* acedidas pelo utilizador, armazenando-as em cache. Este procedimento provoca melhorias a nível da performance da aplicação. Cada *template* encontra-se definida num *state*. Por sua vez, os *states* da aplicação são geridos pela framework de *routing*: *angular-ui-route*.

O elemento “<ion-header-bar>” define o conteúdo do *header* de cada página, neste caso em concreto é inserido o seu título. Por outro lado, o elemento “<ion-content>” é onde se encontra o conteúdo da página propriamente dito.

```
angular.module('exemploIonicApp', ['ionic'])

.controller("MainCtrl",function(){
  console.log("Olá Mundo do MainCtrl");
})

.config(function($stateProvider, $urlRouterProvider){
  $stateProvider

  .state('main', {
    url: "/main",
    templateUrl: "templates/main.html",
    controller: 'MainCtrl'
  })

  .state('page2', {
    url: "/page2",
    templateUrl: "templates/page2.html",
  })

  // estado predefinido
  $urlRouterProvider.otherwise('/main');
});
```

Código 2 – JavaScript de exemplo Ionic.

No que toca ao JavaScript, é inicializado o modulo do angular, referindo o nome da aplicação. Também é feita a importação do módulo “ionic” proveniente da biblioteca. De seguida é criado o controlador principal (*MainCtrl*) onde poderá ser introduzida a sua lógica. O bloco de configuração permite definir vários comportamentos. Neste caso é feita a gestão dos *states*, onde são associados os devidos *templates* e URL, bem como os seus respetivos controladores. Finalmente poderá definir-se um *state* predefinido que é utilizado caso nenhum *state* seja definido aquando exista uma mudança de página.

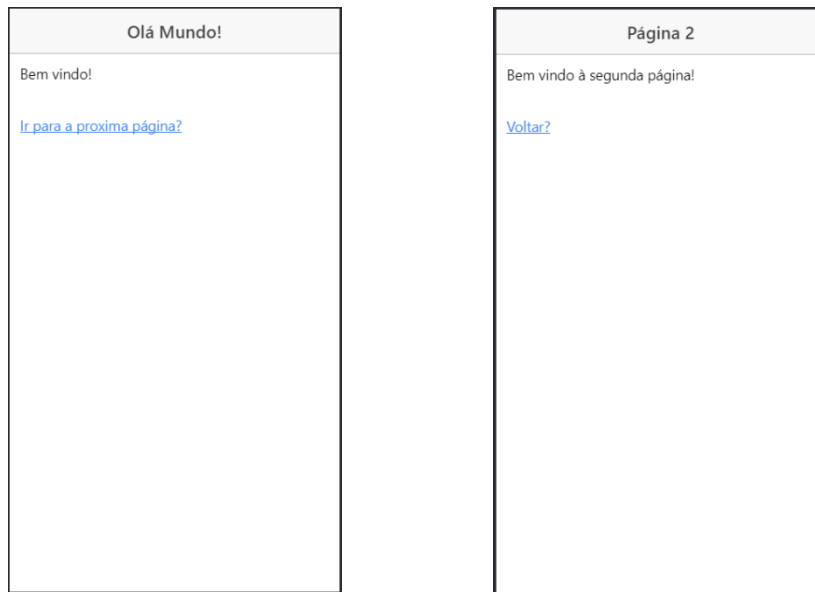


Figura 11 - Hello World em Ionic.



Figura 12 - Logotipo Ionic

3.1.7 JavaScript

O JavaScript (JS) é uma linguagem de programação de alto nível, sendo uma das três principais tecnologias (CSS, HTML e JavaScript) que definem a produção de conteúdos para a World Wide Web[6]. Esta linguagem terá sido originalmente implementada para que scripts pudessem ser executados por parte do cliente sem existir a necessidade deste script passar pelo servidor. Resultando numa comunicação assíncrona que controla o conteúdo do documento apresentado pelo browser. Pode ser considerada assim uma linguagem de programação "segura". Pois não permite acesso de baixo nível à memória ou ao CPU, porque foi inicialmente criado para browsers, que não o exigem.

Atualmente o JavaScript é uma linguagem que suporta múltiplos paradigmas, ou seja, possui características de programação orientada a objetos, imperativa, funcional e baseada em protótipos.

Os seus recursos dependem bastante do ambiente que executa o próprio JavaScript. Por exemplo, o Node.JS suporta funções que suportam JavaScript para ler ou escrever arquivos arbitrários ou executar solicitações de rede.

A linguagem JavaScript permite:

- Adicionar novo HTML à página, alterar o conteúdo existente e modificar estilos de CSS.
- A capacidade de reagir a ações do utilizador, executar em cliques do rato, gerar movimentos do ponteiro e pressionar teclas.
- Envio de informações pela rede para servidores remotos através de pedidos AJAX (*Asynchronous Javascript and XML*). Possibilitando o download e upload de texto ou vários tipos de ficheiros.
- Capacidade de obter *cookies*, construção formulários, criação de mensagens de alertas.
- Suportar o armazenamento local por parte do cliente.



Figura 13 - Logotipo JavaScript

3.1.8 jQuery

O jQuery é considerado uma biblioteca leve (cerca de 263kb), multiplataforma, baseada na linguagem JavaScript. Esta tecnologia foi criada com o objetivo de melhorar a programação em JavaScript para o lado do cliente[12]. Facilitando a construção, estruturação e desenvolvimento dos comportamentos de um documento HTML. Esta biblioteca permite a simplificação do código de tarefas comuns, através da execução de vários métodos, recorrendo apenas a uma única linha de código. Estes métodos executam instruções que necessitariam várias linhas de código JavaScript.

Vantagens da biblioteca jQuery:

- Manipulação do HTML DOM no que toca à seleção de elementos.
- Gestão de eventos de HTML. tais como controle do rato, ou eventos relacionados com elementos como botões ou caixas de texto.
- Manipulação de seletores CSS para elementos que utilizam atributos, tais como *name*, *id* e *classe*, como critério de seleção.
- Uso de efeitos e animações aplicados a elementos presentes na página web.
- Execução de pedidos AJAX
- Objetos do tipo *Deferred* e *Promise* que permitem controlar processamento assíncrono.
- Bastante extensível através da implementação de plugins.
- Compatibilidade com a maior parte dos browsers (Chrome, Firefox, Safari, Opera, Edge e Internet Explorer 9 ou superior)



Figura 14 - Logotipo jQuery

3.1.9 JSON

A tecnologia JSON, ou *JavaScript Object Notation*, é um formato de ficheiro padrão que utiliza uma sintaxe de fácil interpretação pelo humano. Representa um formato de dados utilizado para a comunicação assíncrona entre cliente e servidor sob forma de texto. Tem como objetivo a transmissão de objetos de dados que consistem em pares de atributo-valor e tipos de dados em forma de matriz.

```
{
  "Employees": {
    "Emmployee": [{
      "name": "David",
      "city": "Lisbon",
      "cars": [
        "Ford",
        "VW"
      ],
      "job": "Computer Engineer"
    },
    {
      "name": "Carlos",
      "city": "Porto",
      "cars": [
        "BMW",
        "Seat"
      ],
      "job": "Doctor"
    }
  ]
}
```

Código 3 - Exemplo JSON.

É possível converter qualquer objeto JavaScript em JSON ou qualquer objeto JSON em JavaScript, através das funções *JSON.stringify()* e *JSON.parse()*, respectivamente.



Figura 15 - Logotipo JSON

3.1.10 PHP

O PHP, originalmente denominado *Personal Home Page*, atualmente *PHP: Hypertext Processor*. É uma linguagem de programação desenvolvida principalmente para desenvolvimento para o lado do servidor, de conteúdos web dinâmicos e interativos. É possível embutir código de PHP num documento HTML ou HTML5[10]. A linguagem, permite a utilização funcionalidades através linha de comandos (CLI). A tecnologia PHP é compatível com a maior parte dos sistemas operativos, e de uma forma gratuita. Uma das características mais fortes e significativas do PHP é o suporte para bastantes tecnologias de bases de dados, como o MySQL[10]. O PHP é utilizado em aplicações bastante conhecidas como o Drupal, Facebook, Joomla, WordPress, Magento, FFmpeg e Wikipedia.



Figura 16 - Logotipo PHP

3.1.11 SQL

O SQL (*Structured Query Language*), é uma linguagem de consulta estruturada. O SQL representa a linguagem padrão para armazenamento, manipulação e recuperação de dados para bases de dados relacionais. Vários sistemas de gestão de bases de dados relacionais utilizam SQL, tais como Microsoft SQL Server, Oracle, Ingres, Sybase, Access, etc.

A linguagem SQL permite:

- Criar novas bases de dados.
- Criar novas tabelas em bases de dados.
- Executar consultas específicas numa base de dados.
- Recuperar dados de uma base de dados.

- Criar procedimentos armazenados em bases de dados.
- Criar views numa base de dados.
- Definir permissões para tabelas, procedimentos e views.
- Inserir registros em base de dados.
- Atualizar registros em base de dados.
- Excluir registros de base de dados.

Segue-se um simples exemplo da criação de uma tabela em SQL com informação acerca da localização geográfica de cidades:

- Criação da a estrutura da tabela, definindo os campos, bem como o seu tipo de dados.

```
CREATE TABLE CIDADES
(ID INTEGER PRIMARY KEY,
CIDADE CHAR(20) ,
DISTRITO CHAR(20) ,
LATITUDE REAL,
LONGITUDE REAL) ;
```

Código 4 – Criação da estrutura de Tabela SQL.

- Popular as colunas da tabela anteriormente criada com informações.

```
INSERT INTO CIDADES VALUES (13, 'Tomar', 'Santarém', 40, 8);
INSERT INTO CIDADES VALUES (44, 'Valença', 'Viana do Castelo', 42, 9);
INSERT INTO CIDADES VALUES (66, 'Albufeira', 'Faro', 37, 8);
```

Código 5 - Inserção de valores na tabela.

- Query para mostrar todas as entradas da tabela.

```
SELECT * FROM CIDADES;
```

Código 6 - Seleção a todos os valores da tabela.

ID	CIDADE	DISTRITO	LATITUDE	LONGITUDE
13	Tomar	Santarém	40	8
44	Valença	Viana do Castelo	42	9
66	Albufeira	Faro	37	8

Tabela 2 - Exemplo de tabela criada.

Os projetos trabalhados na duração do estágio utilizavam bases de dados MySQL. Pelo que, conseqüentemente, foi utilizada a linguagem SQL, para gerir valores em base de dados durante a fase de testes.



Figura 17 - Logotipo SQL

3.1.12 XML

A tecnologia XML, ou Extensible Markup Language, representa uma linguagem do tipo marcação, tal como o HTML. O objetivo desta linguagem é definição de regras e contextualização da informação de maneira a construir o documento. No entanto, ao contrário do HTML, esta tecnologia não define como essa informação é mostrada.

```

<Employees>
  <Emmployee>
    <name>David</name>
    <city>Lisbon</city>
    <cars>Ford</cars>
    <cars>VW</cars>
    <job>Computer Engineer</job>
  </Emmployee>
  <Emmployee>
    <name>Carlos</name>
    <city>Porto</city>
    <cars>BMW</cars>
    <cars>Seat</cars>
    <job>Doctor</job>
  </Emmployee>
</Employees>

```

Código 7 – Exemplo XML

Nos ficheiros de configuração, relativos aos projetos baseados nas *frameworks* Cordova e Ionic, é utilizada a linguagem XML por defeito.



Figura 18 - Logotipo XML

3.2 Ferramentas Utilizadas

Neste subcapítulo apresentam-se as ferramentas de desenvolvimento e os equipamentos utilizados durante o estágio, bem como as suas principais finalidades.

3.2.1 Browsers

São ferramentas importantes para realizar os testes iniciais do funcionamento de novas implementações, principalmente a nível de CSS e pedidos ao servidor.

- Apple Safari
- Google Chrome
- Microsoft Internet Explorer
- Microsoft Edge

3.2.2 Editores de texto

Como se trabalhou em várias áreas desenvolvimento, para várias plataformas e em várias máquinas. Foram utilizados bastantes tipos de editores de texto.

- Atom – Principal editor de texto para os projetos. Sendo possível adicionar vários plugins, alterar o tema de cores do código, bem como verificar quais os ficheiros modificados desde a ultima sincronização com o repositório git.
- Android Studio – Suporta compilações baseadas em Gradle. Utilizado para realizar testes nos dispositivos Android, possui um software de monitorização (Android Monitor) que dispõe a leitura de registos da consola (*logs*). Permite a emulação de dispositivos Google (Nexus, AndroidTV, Android Wear).
- Brackets – Editor de texto alternativo para JavaScript, CSS e HTML.
- Eclipse – Utilizado para efetuar a sincronização com o repositório git (*commit*, *push* e *pull*) de cada projeto.
- Notepad++ - Utilizado principalmente para visualizar e editar código de ficheiros independentes, tais como ficheiros de PHP por parte do servidor.
- Visual Studio 2015 – Principal ferramenta de desenvolvimento e testes a nível do Windows Phone. Permite a deteção de erros, implementação de plugins, bem como a emulação de dispositivos Windows Phone com diferentes resoluções e sistemas operativos.
- Xcode – Editor de texto para aplicações iOS, permite fazer *build release* para publicação na Apple App Store. Com o Xcode é possível emular vários dispositivos Apple dependendo da versão desta ferramenta.

3.2.3 Outros

- Paint.net – Ferramenta básica de edição de imagem. Utilizada principalmente para editar ícones com formato de imagem do tipo png e jpeg.
- Putty – É um software gratuito que permite emular um terminal. Suporta vários tipos de protocolos de rede (*SCP, SSH, Telnet, rlogin*). Neste estágio a conexão foi efetuada com *SSH* através de *IP* seguida da introdução do nome de utilizador e password.
- SoapUI – Ferramenta que permite testar facilmente os pedidos REST efetuados a um dado servidor.
- WinSCP – Utilizado para transferir e editar ficheiros armazenados no servidor. Suporta os seguintes protocolos de rede: SFTP, FTP, WebDAV e SCP.
- MySQLWorkbench – Ferramenta para visualizar, criar e editar bases de dados recorrendo à linguagem SQL.

3.2.4 Dispositivos

Foram utilizados os dispositivos móveis disponibilizados pela empresa, bem como dispositivos pessoais de maneira a abranger mais versões de OS. Estes dispositivos são bastante importantes no que toca aos testes a efetuar. Pois quanto maior for o número de dispositivos onde os testes são aplicados com sucesso, maior a confiança de que não existem problemas. Existindo a confirmação que a aplicação poderá ser atualizada.

- Android:
 1. Google LG Nexus 4 (Android 6.0 Marshmallow)
 2. Google LG Nexus 5 (Android 7.1 Nougat)
 3. BQ Aquaris E4.5 (Android 4.4 KitKat)
 4. Samsung Galaxy Tab2 (Android 5.1 Lollipop)

- Windows
 1. Nokia 630 (Windows Phone 8)

- iPhone OS
 1. iPhone 4 (iOS 7.1.2)
 2. iPhone 5 (iOS 9.3.5)

4 Metodologia

Neste capítulo apresentar-se-ão os diferentes projetos trabalhados ao longo do estágio. Demonstrando as suas funcionalidades e serviços, seguidos da sua respetiva metodologia de desenvolvimento.

4.1 Projeto – Sou Cidadão

O Sou Cidadão foi a primeira aplicação móvel desenvolvida pelo Grupo Compta, onde a ideia terá surgido. O seu objetivo passa pela aproximação dos cidadãos às autarquias e freguesias nacionais, criando uma interface digital destinado à partilha de informações. Deste modo, o contacto é facilitado e permite, de forma interativa, o acesso a vários serviços, tais como, visualização de notícias ou eventos, registo de ocorrências ou intervenções, registo de animais e pedidos de recolha de monos e verdes. Posteriormente procedendo à solicitação direta destes serviços aos seus respetivos órgãos de gestão. A aplicação recorre a ferramentas avançadas que recorrem às capacidades do dispositivo, como a georreferenciação e o acesso à sua câmara, que permitem uma gestão mais eficaz da edilidade, melhorando sua relação com os cidadãos.

O Sou Cidadão é um projeto ambicioso, que resultou numa aplicação ainda em crescimento. Possuindo potencial para operar em bastantes entidades camarárias a nível nacional, bem como proceder à implementação de uma maior abrangência de serviços.



Figura 19 - Logotipo Sou Cidadão

Este projeto recorre ao sistema Drupal. Este atua como plataforma de *backoffice* para os seus clientes, fornecendo acesso limitado ao seu respetivo município. Este sistema está encarregue de organizar e gerir qualquer tipo de conteúdos criados e adicionados à base de dados da aplicação. É totalmente configurável por parte dos desenvolvedores do projeto, através da sua autenticação administrativa. Os WebService, desenvolvidos em linguagem *PHP*, gerem a comunicação de ambas as interfaces, Drupal e aplicação móvel, com o servidor de base de dados. A troca de mensagens entre as interfaces do Sou Cidadão e os seus WebService é estabelecida através da plataforma RabbitMQ, efetuado pedidos do tipo SOAP.

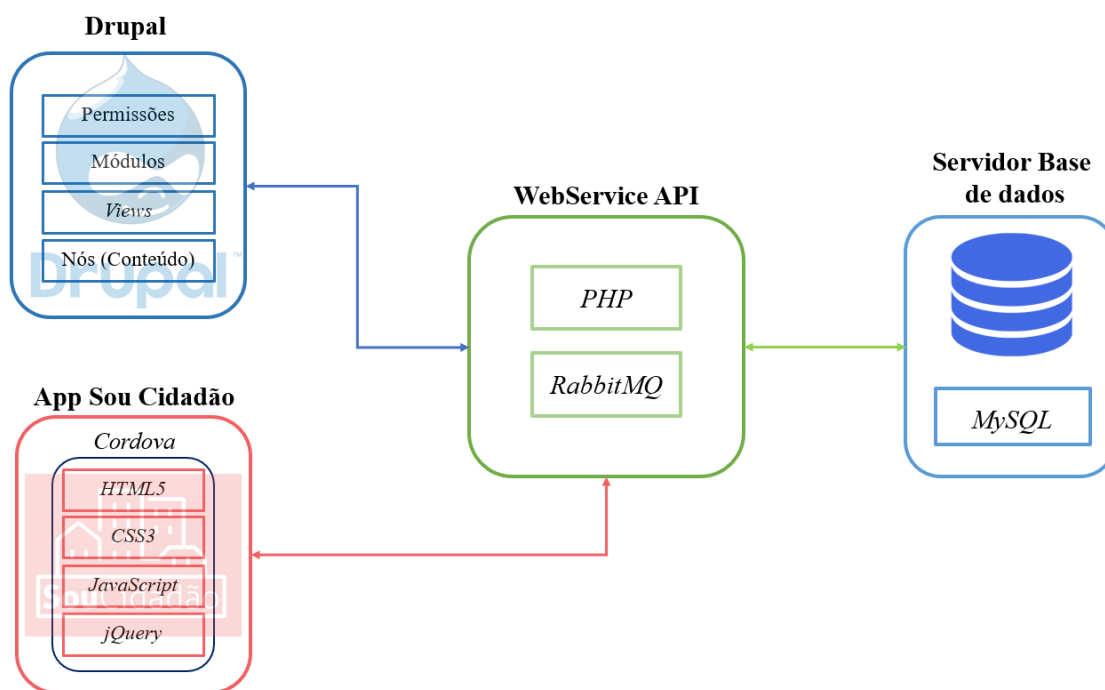


Figura 20 - Arquitetura do Projeto Sou Cidadão

No que toca à estrutura da aplicação móvel, esta é composta por um menu que permite a seleção dos serviços principais (Notícias, Eventos e Ocorrências) na parte superior do ecrã. É ainda possível aceder ao menu lateral, que contém serviços específicos a cada freguesia ou autarquia, bem como informações uteis acerca da mesma, como contactos e moradas de entidades específicas.

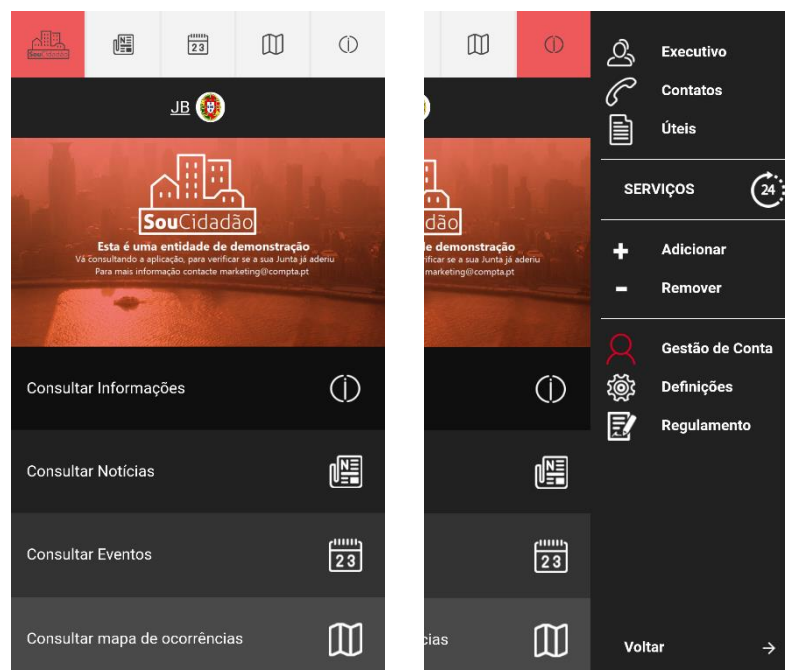


Figura 21 - Ecrã inicial e menu lateral.

Os próximos subcapítulos descrevem os serviços ou funcionalidades que foram trabalhadas e desenvolvidas no projeto Sou Cidadão. De notar ainda, que como forma de coerência contextual para a leitura, o cidadão será referenciado como “utilizador” e a junta de freguesia ou autarquia será referenciada como “entidade”.

4.1.1 Serviço – Ocorrências

Permite ao utilizador enviar sugestões ou reclamações, que necessitem da intervenção direta da Câmara Municipal ou Junta de Freguesia, acerca da qualidade ou estado dos serviços e espaços públicos.

O registo da ocorrência é inicializado no menu “Ocorrências”, onde é apresentado um mapa da localidade em questão. De seguida, pressionar o botão “Reportar Ocorrência” onde se poderá introduzir variadas informações, de notar a seleção da localização exata através do marcador introduzido no mapa, bem como o envio da fotografia, por exemplo do antes e depois.

No que toca ao mapa de ocorrências, existia um problema relacionado com as imagens atribuídas ao marcador responsável pela representação de várias ocorrências. Quando se aplica um *zoom out* no mapa de ocorrências, de maneira a não surgirem imagens sobrepostas, o plugin (googlemaps/js-marker-clusterer) responsável pelos marcadores cria um marcador com um contador de itens que agrupou. Estes agrupadores de marcadores não possuíam nenhuma imagem atribuída. O *plugin* está preparado para aplicar cinco tipos de imagens referentes a cinco níveis de agrupamento. Pelo que terão sido criadas imagens, em contexto com os marcadores existentes, que definem estes cinco níveis. Garantindo ainda que o número que representa o contador de agrupamento permaneça centrado.



Figura 22 - Imagens marcadores de agrupamento.



Figura 23 - Exemplo agrupamento de marcadores.

4.1.2 Serviço - Orçamento Participativo

Este serviço tem como objetivo reforçar a participação cívica na governação de cada freguesia, disponibilizando aos utilizadores, acesso ao Orçamento Participativo (OP):

Uma campanha de Orçamento Participativo segue uma sequência de seis fases distintas, são elas:

- Preparação de novo ciclo (1) – O utilizador apenas poderá verificar a existência de uma nova campanha, e a sua respetiva informação (nome, data, descrição, valor do orçamento e fase da campanha)
- Recolha de Sugestões (2) – Nesta fase é possível ao utilizador aceder ao menu das sugestões, existindo a possibilidade de criar ou comentar sugestões existentes.
- Análise técnica das Sugestões (3) – A criação de novas sugestões é bloqueada, pois está a ocorrer uma análise por parte da entidade responsável.
- Votação dos projetos (4) – O menu das sugestões passa a estar bloqueado, enquanto que o menu de visualização de propostas é ativado.
- Apresentação pública dos resultados (5) – Nesta fase ambos os botões de sugestões e votação se encontram desativados.
- Avaliação do processo e elaboração de relatório final (6) – Tal como na fase anterior, ambos os botões de sugestões e votação se encontram desativados.

A tarefa a efetuar neste serviço é a implementação dos comportamentos dos botões relativamente às fases de Orçamento Participativo.

Foi criada uma classe CSS com o objetivo de desativar a possível interação com os botões ou elementos pretendidos:

```

.disabled {
  opacity:0.3;
  touch-action: none;
  ms-touch-select: none;
  pointer-events:none;
  -webkit-transition-property: none !important;
  transition-property: none !important;
  -webkit-animation: none !important;
  animation: none !important;
  /*iOS*/
  -webkit-touch-callout: none;
}

```

Código 8 - CSS referente à classe disabled.

Esta classe de CSS (*disabled*) retira alguma opacidade ao elemento para dar o efeito de “botão bloqueado”. Incluí ainda algumas propriedades com o objetivo de desativar o “tap/touch/swipe” no elemento, bem como o seu efeito de *feedback*.

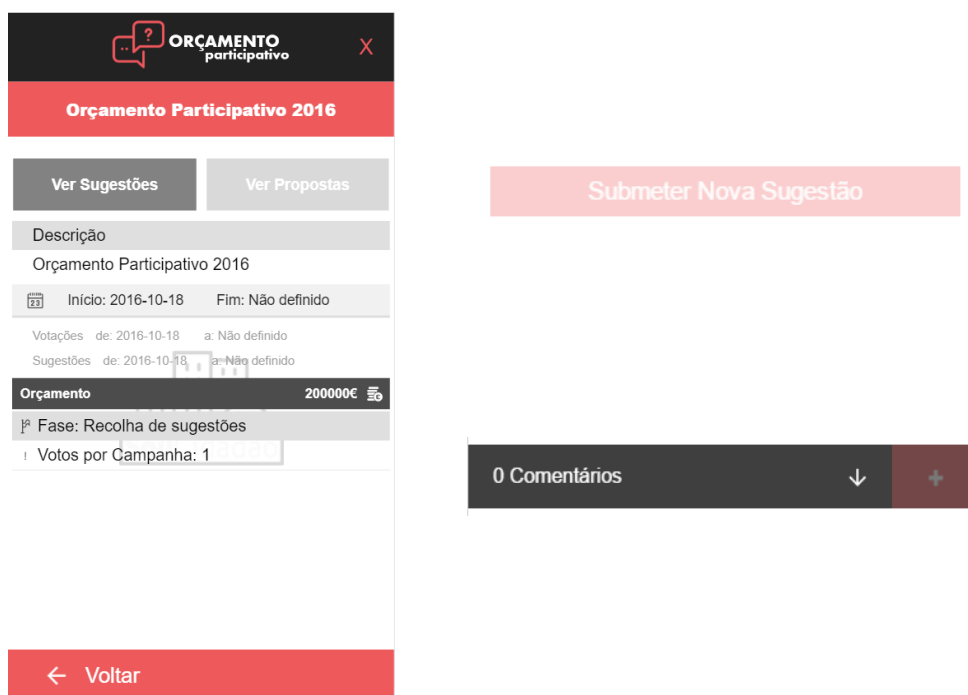


Figura 24 - Exemplos de elementos desativados de acordo com a fase OP.

No que toca ao sistema de propostas, disponível na fase de “Votação dos projetos (4)”, terá sido implementado um mecanismo de limitação de votos. Estes votos são efetuados mediante a seleção de dadas *checkboxes* para cada proposta. A ideia implementada consiste na desativação da proposta/*checkbox* de acordo com o número máximo de votos. Sendo novamente reativada caso o utilizador remova um voto já inserido. Recorreu-se novamente à classe anteriormente criada (*disabled*) para este efeito.

Proposta	Voto
Requalificação do pa...	☒
requalificação parqu...	☒
Beneficiação da Rua...	☐
Instalação de novos ...	☒

Votar

Figura 25 - Exemplo de boletim de voto para um máximo de três votos.

4.1.3 Serviço – Recolha de Monos e Verdes

Criação de serviço destinado à recolha de objetos volumosos e fora de uso, bem como lixo biológico de grandes dimensões. A recolha é efetuada por parte dos serviços camarários de cada entidade.

Na aplicação, terá sido adicionada ao menu lateral de opções de serviços, uma nova opção com a respetiva imagem referente ao serviço em questão. Esta opção encaminha o utilizador para uma nova página com dois botões que definem o tipo de recolha a efetuar (Monos ou Verdes). Cada botão encaminha o utilizador a uma página idêntica a ambos os tipos, onde lhe é solicitada informação relativa ao pedido. Está presente uma tabela com os horários das recolhas relativas à entidade em questão. Para além das informações de contacto, é solicitado ao utilizador a data, materiais pretendidos, localização e

fotografia(s). Os campos de *input* referentes a estes dados, foram criados respeitando o CSS dos restantes serviços da aplicação. Sendo que as respetivas validações destes *inputs* foram revistas e reajustadas de acordo com o pretendido.

The screenshot shows the 'Recolha Monos e Verdes' application interface. The left sidebar contains navigation options: 'O. Participativo', 'Lic. Animais', 'Recolha de Monos e Verdes', 'Referendo Local', and 'Informação Transportes'. The central content area features two large red buttons labeled 'Monos' and 'Verdes'. The right sidebar contains a form for submitting a collection request, with fields for 'Data do pedido de recolha', 'Data (obrigatório)', 'Materiais', 'Materiais (obrigatório)', 'Localização', 'Localização (obrigatório)', 'Observações', 'Observações', 'Imagens adicionadas', and a 'Submeter' button. A table titled 'Horários do Circuito' is also visible, showing days and zones.

Dia	Hora	Zona
Segunda-feira	— —	
Terça-feira	— —	
Quarta-feira	— —	
Quinta-feira	— —	
Sexta-feira	— —	

Figura 26 - Ecrãs de recolha de monos e verdes.

4.1.4 Notificações

No sistema de notificações do Sou Cidadão, o utilizador deverá ser notificado aquando da criação ou alteração de certos conteúdos na base de dados da aplicação. Nomeadamente quando é criada uma notícia, um evento, uma alteração ao estado de uma ocorrência e ainda caso seja criado um novo comentário a uma dada ocorrência.

As notificações apenas são recebidas caso o utilizador assim o desejar. O utilizador deverá seleccionar a(s) *checkbox(es)* relativas às notificações aquando da adição da entidade pretendida. No caso das ocorrências, o utilizador deverá acompanhar a ocorrência pretendida de maneira a receber notificações do seu estado, bem como novos comentários.

Ao seleccionar estas opções de notificações, é enviado para a tabela *push_devices* da base de dados do servidor, o ID do dispositivo bem como o seu *token*. Deste modo, o servidor saberá para quais dispositivos enviar as *push notifications*.

Do lado do servidor, ao receber as informações relativas à notificação a ser efetuada, estas são tratadas consoante o seu tipo, e encaminhadas para os dispositivos. Estes dividem-se pelos 3 tipos de plataformas suportadas pela aplicação (Android, iOS e Windows Phone).

Para cada plataforma existe um serviço diferente de envio das *push notifications*:

- **GCM** – Google Cloud Messaging, que trata do envio das notificações para a plataforma Android.
- **APNS** – Apple Push Notification Service, que trata do envio das notificações para a plataforma iOS.
- **MPNS** – Microsoft Push Notification Service, que trata do envio das notificações para a plataforma Windows Phone.

A aplicação está preparada para receber as notificações, no entanto a sua informação não é tratada. Ou seja, a notificação é recebida e quando o utilizador pressiona a notificação com a aplicação em background, esta é inicializada normalmente mostrando a página inicial.

Foram propostos melhoramentos significativos ao nível destas notificações. Começando pela reestruturação do corpo da notificação, e implementação de um mecanismo de encaminhamento automático para a respetiva página de conteúdo. Por exemplo, no caso de um utilizador receber uma notificação acerca de uma nova notícia, ao pressionar essa notificação é reencaminhado automaticamente para página dessa mesma notícia.

Adicionalmente, as notificações deverão mostrar, caso seja possível, a imagem referente ao conteúdo publicado, bem como o seu título e mensagem.

a) Desenvolvimento WebService.

Por parte do servidor são enviadas as seguintes variáveis:

- *nid* – é atribuído o *nid* (node ID) da entidade.
- *pushid* – é atribuído o id do conteúdo da publicação (notícia, evento, ocorrência e comentário da ocorrência).
- *imagem* – é atribuído o URL da imagem.
- *texto* – é atribuído o nome da entidade.
- *titulo* – é atribuído o título da publicação.
- *tipo* – é atribuído o tipo de publicação.

Para além das variáveis inicialmente existentes, adicionaram-se as variáveis *pushid* e *imagem*. Estas são igualmente encaminhadas pelo método *MQsendPush*, e

posteriormente, pelos serviços de envio das *push* de cada uma das plataformas, presentes no método *sendPush*. Estas alterações foram efetuadas diretamente no ficheiro *PHP* (*WS_drupal_sendPush.php*), encarregue de gerir o envio das *push notifications*.

A secção *WebService* na plataforma *Drupal* referente ao método *MQsendPush*, também necessita de ser atualizada. Esta atualização é efetuada de acordo com as alterações efetuadas no *PHP*, tal como se pode verificar na próxima figura.

[Início](#) » [Administração](#) » [Configuração](#) » [Web services](#) » [Web service descriptions](#) » [sendPush](#)

Edit operation

Etiqueta *
MQsendPush
The human-readable name of the operation.

Nome *
MQsendPush
The machine-readable name of this operation is used internally to identify the operation.

Parameters
Specify the parameters for the operation. For each parameter you have to specify a certain data type and a unique name containing only alphanumeric characters and underscores. You can also specify a default value for the parameter and if it is required.

[Mostrar coluna de pesos](#)

Tipo de dados	Múltiplo	Nome	Valor predefinido	Obrigatório
texto	☐	nid		☒
texto	☐	pushid		☒
texto	☐	imagem		☒
texto	☐	texto		☒
texto	☐	titulo		☒
texto	☐	tipo		☒
--	☐			☒
--	☐			☒
--	☐			☒

Figura 27 – Configuração de variáveis do método *MQsendPush* no *Drupal*.

Ainda no servidor, referente ao ficheiro para as tarefas *SOAP* (*soap_tasks.js*). As novas variáveis *pushid* e *imagem*, também foram adicionadas ao método *process*, especificamente dentro do objeto de informações relativo à tarefa do tipo *push*.

b) Desenvolvimento Aplicação.

Ao nível da aplicação é utilizado um plugin Cordova (*PushPlugin*) com a versão 2.4.0. Este plugin tem como objetivo gerir a receção de notificações para diferentes plataformas (Amazon Fire OS, Android, iOS, Windows Phone e Windows8).

No que toca ao tratamento da informação por parte da aplicação, este necessita de ser igualmente diferenciado pelas três plataformas utilizadas. Existindo um método para cada uma delas (*onNotificationGCM*, *onNotificationAPN*, *onNotificationWP8*).

```
function push_initialize()
{
var pushNotification = window.plugins.pushNotification;
// Platform-specific registrations.
if (app_deviceType == 'android'){
// Register with GCM for Android apps.
pushNotification.register(successHandler, errorHandler, {"senderID": GCM_SENDER_ID,
"ecb": "onNotificationGCM"});
}
else if (app_deviceType == 'ios'){
// Register with APNS for iOS apps.
pushNotification.register(tokenHandler, errorHandler, {"badge": "true",
"sound": "true",
"alert": "true",
"ecb": "onNotificationAPN"});
}
else if (app_deviceType == "win32nt"){
// Register with MPNS for WP8 apps.
pushNotification.register(channelHandler, errorHandler, {"channelName": "MyPushChannel",
"ecb": "onNotificationWP8",
"uccb": "channelHandler", "errcb": "ErrorHandler"});
}
}
```

Código 9 - Função push com base no tipo de plataforma.

Ao receber a notificação, é necessário garantir que a aplicação já possui a informação relativa ao conteúdo publicado, para que desta forma este possa ser devidamente mostrado. Inicialmente armazena-se o *pushid* na *localStorage*, de maneira a que possa ser posteriormente acedido pelos métodos relativos a cada tipo de conteúdo. De seguida verifica-se se o utilizador realmente possui a entidade em questão. Finalmente, é feito o pedido ao servidor da informação pretendida e mostrado a publicação relativa ao *pushid* recebido.

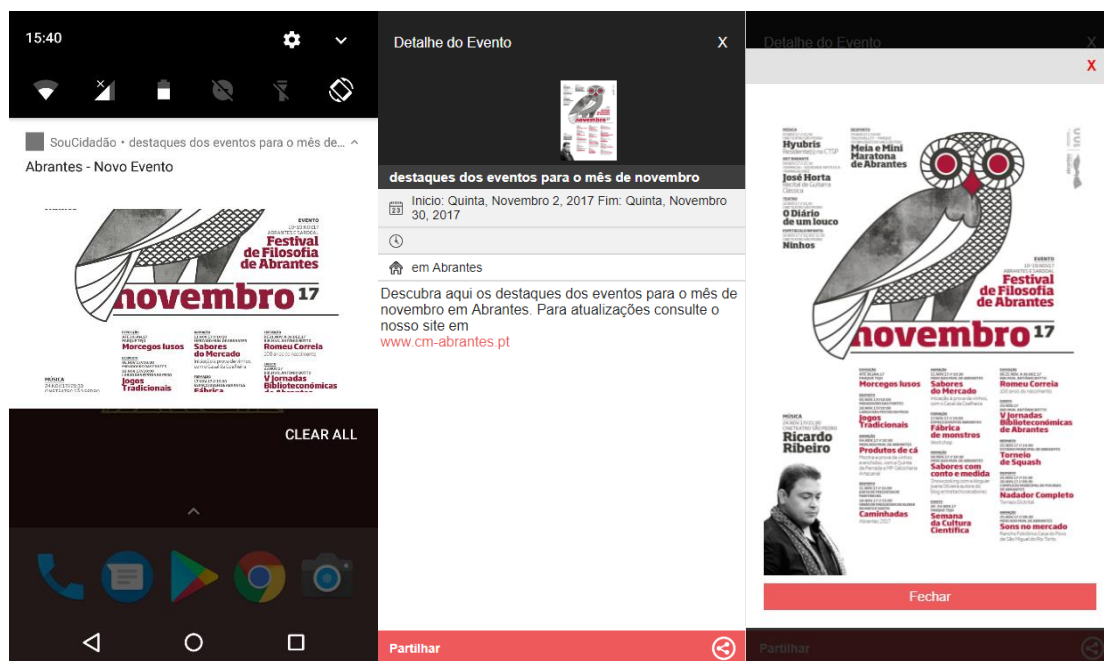


Figura 28 - Exemplo de notificação.

4.2 Projeto – PaxVoice

Trata-se de um projeto submetido em 2015, por parte da plataforma nacional de inovação do Grupo Compta, chamada Lusídeias. Tendo sido selecionado como uma das aplicações a desenvolver ao abrigo deste programa.

A aplicação móvel PaxVoice fornece aos seus utilizadores dados estatísticos de opinião pública relativamente aos mais variados assuntos da atualidade, proporcionando uma experiência interativa bastante simples e intuitiva, com dados por idade, género e localização. Encontra-se disponível nas plataformas móveis Android, iOS e Windows Phone, sendo também possível aceder ao PaxVoice através do seu Website.

O PaxVoice terá sido publicado com a expressão “*Power to the People*”, acentuando a missão deste projeto: fazer com que a opinião das pessoas conte, de maneira influenciar a ação dos decisores competentes.

Ainda de notar que esta aplicação obteve sucesso imediato, com a sua incorporação no programa televisivo do canal RTP, “Prós e Contras”. Onde é utilizada para obter a opinião do público e telespetadores acerca de determinados temas diários.

A aplicação foi desenvolvida e estruturada com base na *framework* Ionic, possuindo uma integração com a *framework* AngularJS. Ao contrário do projeto anteriormente apresentado, não houve interação direta com os desenvolvimentos efetuados ao nível dos WebService. Tendo os desenvolvimentos incidido maioritariamente sobre o desenvolvimento da aplicação, com alguma interação sobre a base de dados.

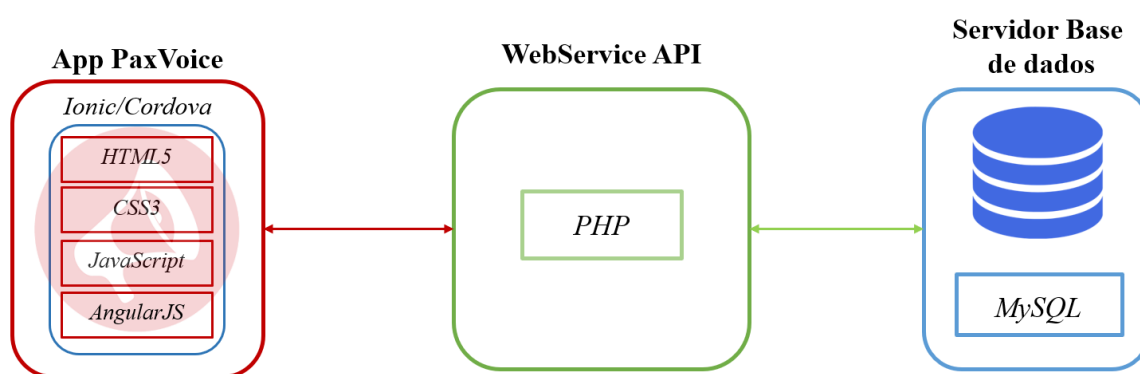


Figura 29 - Arquitetura do projeto PaxVoice

Originalmente, o sistema de registo da aplicação era apenas feito através da rede social Facebook, contando com uma integração com a rede social Twitter. Sendo que o desafio proposto durante o estágio, fosse a implementação de um novo sistema de registo e autenticação através de email e password.

Uma vez efetuado o login, o utilizador poderá criar uma voice (opinião) ou participar de forma confidencial nas voices de outros utilizadores. No que toca à criação de uma voice, o utilizador pode recorrer até 140 caracteres. Cada voice poderá receber votos positivos ou negativos. Também é possível outros utilizadores inserirem comentários ou sustentar a sua opinião.

No ecrã da voice em questão, é possível aceder a diversas informações e estatísticas relevantes, tais como dados por idade, género e nacionalidade. O utilizador pode ainda integrar a sua conta Twitter e criar voices automaticamente através de um tweet, adicionando *#paxvoice*.



Figura 30 - Logotipo PaxVoice

Os próximos subcapítulos descrevem os serviços ou funcionalidades que foram trabalhadas e desenvolvidas relativamente ao projeto PaxVoice.

4.2.1 Serviço – Registo e Login por Email

Como foi referido anteriormente, de maneira a expandir as possibilidades de acesso à aplicação, visto que nem todos os possíveis utilizadores integram a rede social Facebook. Terá sido proposta a tarefa de implementação de um sistema de login através de conta de email e password. Completando este serviço com os mecanismos de verificação de email e reposição de password.

A implementação deste serviço requer a criação ou alteração de uma série de páginas (*views*). Nas próximas alíneas, segue-se uma apresentação dos novos ecrãs implementados no PaxVoice, seguidos da sua respetiva descrição e metodologia de desenvolvimento:

- a) Ecrã Inicial - De maneira a diferenciar o novo tipo de autenticação do método original, foi adicionado um novo botão (“Login”) no ecrã inicial. Possuindo *labels* que indicam as suas possíveis opções.

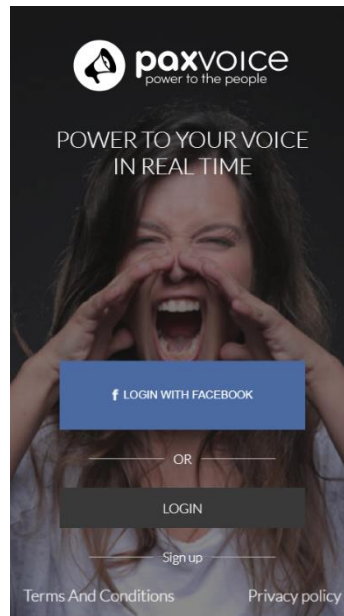


Figura 31 - Ecrã inicial PaxVoice.

Visto que irão ser efetuados bastantes pedidos ao servidor nas seguintes implementações, foi criada uma janela de alerta recorrendo à biblioteca Angular Material. De maneira a ser facilmente acedida por todos os controladores, terá sido criado um serviço AngularJS com as suas configurações. Esta janela avisa o utilizador caso o dispositivo não possua ligação à Internet. Possui um botão “OK” que ao ser pressionado encaminha sempre o utilizador para o ecrã inicial.

- b) Ecrã Autenticação por Email – Através desta página, o utilizador tem acesso a todas as opções relacionadas com a autenticação por email: login, registo, recuperação de password, e reenvio da confirmação de email. Alterou-se a imagem de fundo, e foram criadas novas classes de CSS para os novos campos de *input* bem como as suas respetivas *labels*.

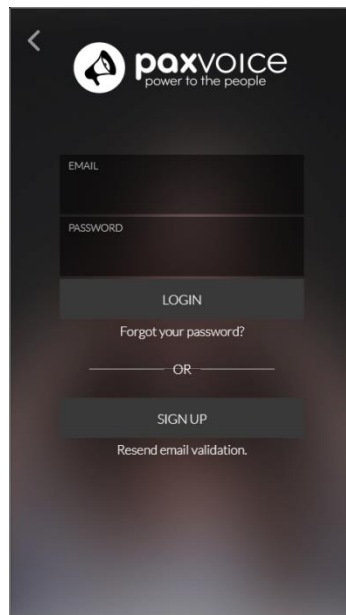


Figura 32 - Ecra de autenticação.

No que toca ao login, foram implementados sistemas de validação para este sistema.

A framework Ionic aliada ao AngularJS, permite efetuar uma verificação em tempo real do *scope*, ou seja, é possível validar dados introduzidos sem existir nenhuma ação por parte do utilizador, por exemplo, pressionar um botão. Pelo que não é efetuado nenhum pedido ao servidor caso pelo menos um dos *inputs* não seja preenchido, ou introduzido incorretamente. Na hipótese de a validação inicial ser efetuada com sucesso. A informação introduzida pelo utilizador nos *inputs* relativos ao email e password, é convertida para base 64. Sendo posteriormente encriptados com base numa biblioteca de encriptação específica a todo o projeto. Gerando assim uma chave de acesso (*access token*) que é utilizada para realizar o pedido (*http.get*) ao servidor. A aplicação atua de acordo com o tipo de resposta obtido. No caso de o utilizador não existir ou a password esteja errada, é mostrado um alerta a referir este erro. Caso o email introduzido não esteja verificado, também será mostrada um alerta idêntico, referindo esse mesmo problema. Finalmente, caso os dados introduzidos sejam devidamente validados com sucesso, é mostrado uma animação (*gif*) de carregamento, de seguida encaminhando para o menu de utilizador. Os dados de sessão são armazenados localmente com as seguintes variáveis: ID de utilizador (*userid*); chave de acesso (*token*); nome da imagem de

utilizador (*userimage*); tipo de autenticação, ou seja, se pertence à rede Facebook ou não (*isFB*); email de utilizador (*email*).

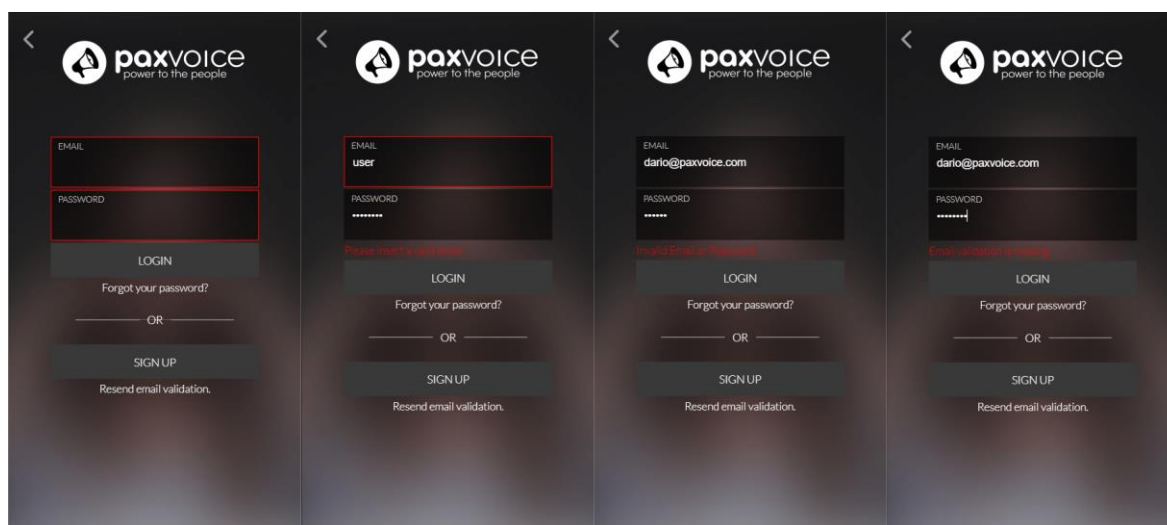


Figura 33. - Erros de validação no processo de autenticação

Os restantes elementos do ecrã de login, encaminham o utilizador para os outros serviços relacionados com a autenticação, e encontram-se referidos nas alíneas seguintes.

- c) Ecrãs Registo – Com o objetivo de estruturar e organizar o registo no PaxVoice, este processo terá sido dividido por duas fases. Como tal são criados dois controladores (*RegistCtrl* e *RegistCtrl2*) para cada um dos ecrãs. Deste modo existem dois *scopes* distintos. Pelo que é necessário partilhar a informação entre eles, de maneira a concluir o registo sem haver perdas de informação. Para ultrapassar este problema, foi criado um serviço (*service*) em AngularJS com um objeto que armazena os dados de registo. Ao inserir esse serviço em ambos os controladores, é possível aceder ao serviço previamente criado. Partilhando assim a informação contida no objeto, entre os elementos dos dois ecrãs de registo.

No primeiro ecrã de registo encontram-se os *inputs* referentes à informação a ser utilizada na autenticação. Novamente, os dados introduzidos são validados em tempo real. Sendo que é efetuada uma verificação de conformidade entre as duas palavras-chave, e onde também é assegurado o seu nível de segurança. Isto sem haver necessidade de pressionar o botão “*continue*”. Após ocorrer a validação com sucesso da informação relativa à autenticação, é efetuado um pedido ao servidor com os dados encriptados. Para este pedido, o único erro que poderá surgir é caso o email introduzido para registo já exista em base de dados, ocorrendo um aviso e impossibilitando o avanço.

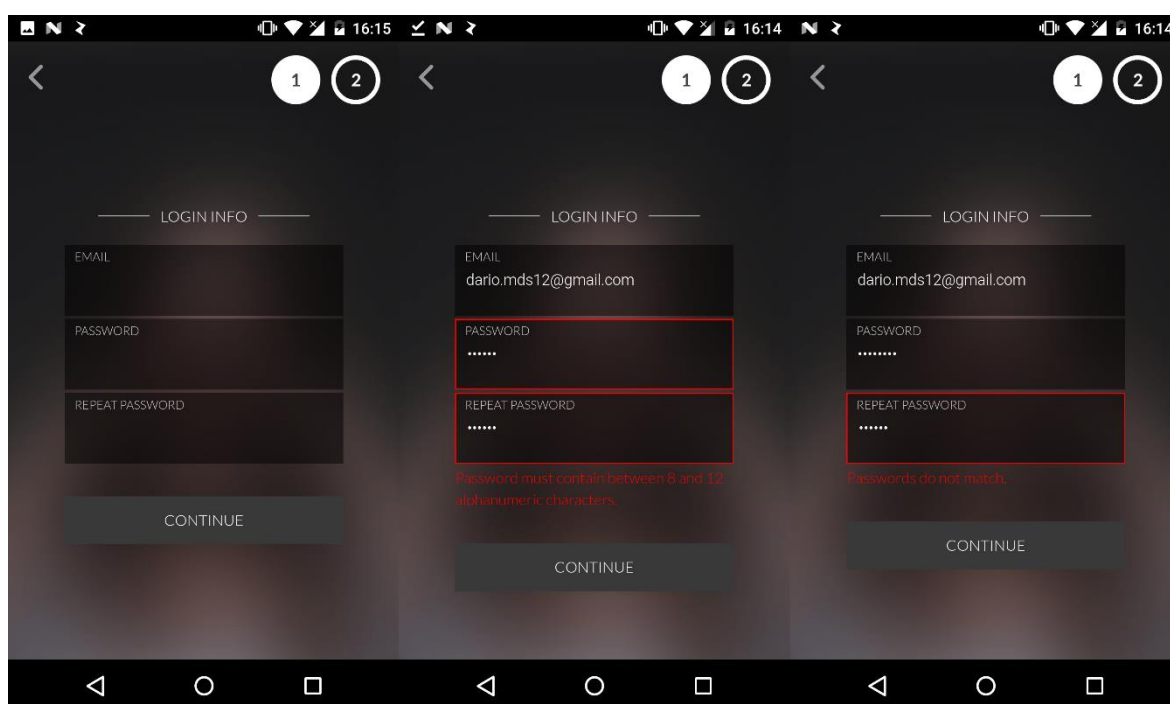


Figura 34 - Erros de validação no processo de registo.

Quando a informação introduzida é válida, o utilizador é encaminhado para o segundo ecrã de registo, onde poderá colocar as suas informações pessoais:

- Fotografia (não obrigatório)
- Nome (obrigatório)
- Género (não obrigatório)
- Data de nascimento (obrigatório)

- País (obrigatório)
- Distrito (não obrigatório)
- Cidade (obrigatório)

A inserção da imagem de utilizador é iniciada ao pressionar o ícone circular com o avatar genérico. Surgem dois novos botões quando este ícone é pressionado, os botões representam duas formas distintas para introduzir uma imagem de utilizador. O botão “camera” irá inicializar a câmara do dispositivo. Ao passo que o botão “gallery” irá aceder à memória do dispositivo, abrindo a galeria de imagens, onde o utilizador poderá escolher a fotografia pretendida. A imagem selecionada tanto pelo método da câmara, como pela memória interna, poderá ser redimensionada recorrendo ao software da plataforma em questão. Mantendo uma relação de aspeto de 1:1. A imagem é então convertida para base64 recorrendo ao plugin *cordovaCamera*, sendo posteriormente guardadas com o formato png na máquina do servidor aquando do registo.

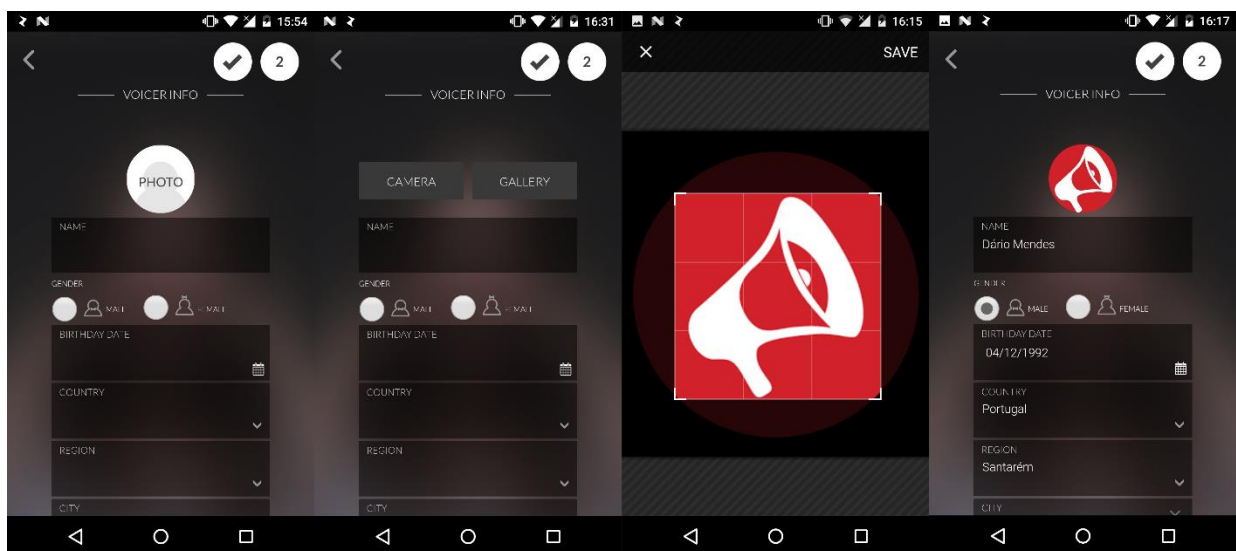


Figura 35 - Processo de introdução de imagem no registo.

Relativamente ao *input* para o campo nome, para além da validação de campo obrigatório, foi implementada uma proteção a nível de segurança que impossibilita a inserção de caracteres especiais. Esta proteção terá sido implementada recorrendo à

criação de uma diretiva AngularJS, que automaticamente elimina qualquer caráter especial que seja introduzido.

No que toca à inserção da data, esta encontra-se limitada com valores mínimos e máximos. Devido a problemas com a definição desses valores mínimos e máximos em algumas plataformas, foi criada uma diretiva AngularJS para este efeito. Dependendo do sistema operativo do dispositivo, é aberto um *popup* que permite efetuar a seleção da data. No caso da plataforma WP8 foi utilizado a biblioteca Angular Material (*md-datepicker*) para este efeito. Pois a versão do sistema operativo do dispositivo existente, não possuía um *popup* automático.

Visto que uma das vertentes da aplicação é a recolha de dados acerca do país. Sendo posteriormente organizados em forma de gráfico percentual, relativamente aos votos efetuados em cada voice. Torna-se necessário que o utilizador introduza o seu país aquando do registo. Os campos dedicados ao preenchimento do país e região são do tipo *select*. A lista de países é fornecida através de um pedido efetuado ao servidor, retornando um objeto com a lista atualizada. Assim que o utilizador seleciona um país, caso este país possua distritos, ou “*region*”, é efetuado um novo pedido ao servidor para obter essas regiões. No que toca à inserção da cidade, esta é realizada de forma diferente. O seu campo é do tipo *autocomplete*, recorreu-se à biblioteca Angular Material (*md-autocomplete*) para este efeito. A pesquisa é efetuada assim que o utilizador insira três ou mais caracteres, retornando uma lista de cidades com base nesses caracteres e o país anteriormente selecionado.

Assim que as informações são introduzidas e validadas, o utilizador poderá pressionar o botão “sign up” de maneira a confirmar o seu registo. A interação com o servidor é inicializada pelo upload (*http.post*) da imagem anteriormente selecionada. Quando esta é carregada com sucesso, o nome da imagem gerado na base de dados é devolvido pelo servidor. Este nome é adicionado ao objeto “*user*” que de seguida é utilizado para realizar o pedido (*http.post*) referente ao registo. Assim que o pedido é concluído com sucesso, será mostrado um alerta de confirmação de registo, avisando o utilizador que deverá efetuar a confirmação do seu email.

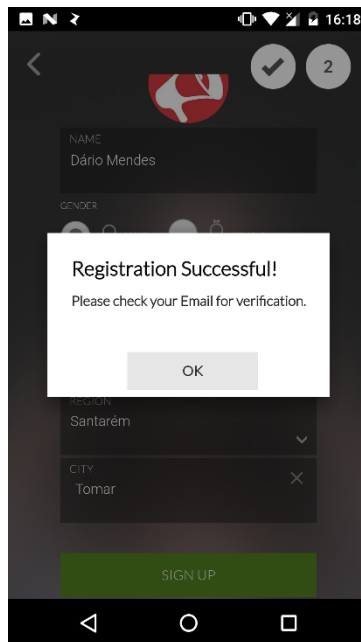


Figura 36 - Ecrã de registo efetuado com sucesso.

- d) Ecrã Recuperação Password – Trata-se de um ecrã simples que permite efetuar a reposição da password para uma dada conta de utilizador. Para este efeito, basta introduzir o email pretendido e proceder à sua confirmação.

Tal como os restantes tipos de *input*, este campo de inserção de email também possui a sua validação. Destacando o caso de o email introduzido, não estar registado na aplicação, impossibilitando o envio. Caso todas as validações sejam efetuadas e o pedido realizado com sucesso, o utilizador é notificado com um alerta de confirmação. Seguindo a hiperligação fornecida pelo email recebido, o utilizador é encaminhado para o website do PaxVoice onde poderá repor a sua password.

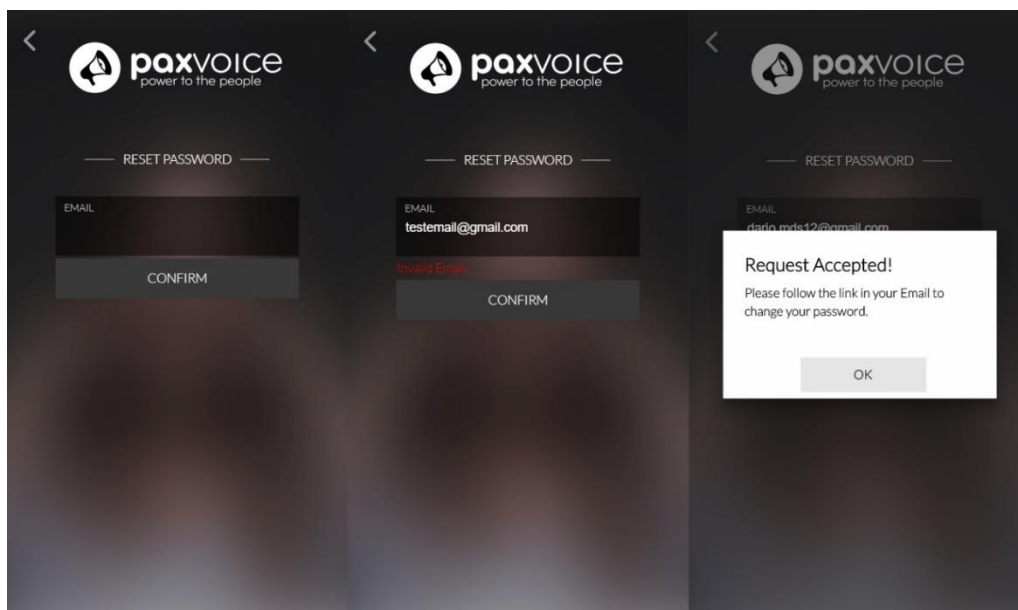


Figura 37 - Ecrã de pedido de reposição de password.

- e) Ecrã Reenvio de Email de Verificação – Idêntico ao ecrã apresentado na alínea anterior. Este ecrã permite efetuar um pedido de reenvio de email de verificação. Possuindo as mesmas características e comportamentos, apenas diferindo no email recebido pelo utilizador. A hiperligação recebida encaminha o utilizador para o website do PaxVoice, concretamente, uma página de confirmação da verificação de email.

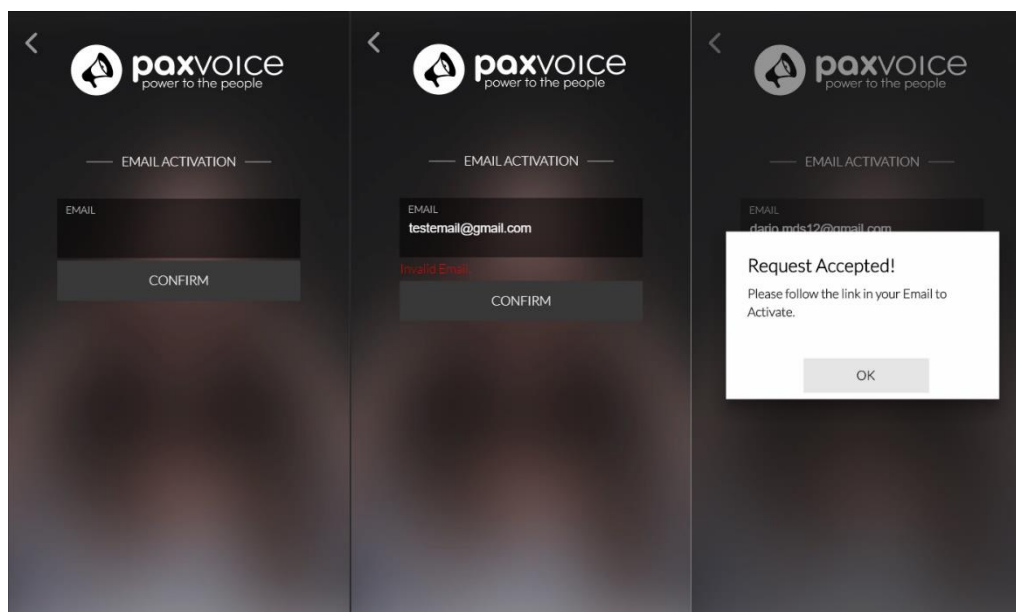


Figura 38 - Ecrã de confirmação de Email.

- f) Ecrã Menu Inicial – Este é o ecrã que surge imediatamente após efetuada a autenticação com sucesso. Através deste ecrã é possível ao utilizador aceder a todas as funcionalidades da aplicação (Home, Trendy, Fresh, My Votes, User e Settings). Relativamente a este ecrã foram corrigidos alguns problemas que surgiram devido à alteração do serviço das definições de utilizador, referido no próximo subcapítulo.

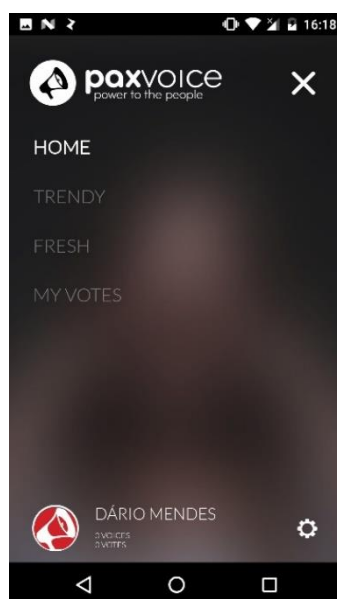


Figura 39 - Ecrã Menu após login.

Mediante a alteração de alguma informação por parte utilizador no que toca aos seus dados no ecrã definições. Mais concretamente, o seu nome, imagem e com especial atenção à sua password, pois altera diretamente o *access token*. Os dados de sessão para este ecrã (menu) não eram atualizados. Sendo necessário alterar o comportamento a nível do controlador deste ecrã.

Implementou-se um *scope watch*, função AngularJS que incide sobre um dado objeto presente na variável *scope* do controlador. Neste caso sobre o objeto “*userdata*” que contem as informações acerca do utilizador autenticado. Deste modo, quando é detetada alguma alteração na variável “*userdata*”, as variáveis *scope* relativas à informação mostrada neste ecrã (*username* e *imageUrl*) são devidamente atualizadas, resolvendo o problema de sincronismo de dados.

4.2.2 Serviço – Definições de Utilizador

Com a implementação do serviço de registo através de email e password, a informação relativa ao utilizador passa a ser introduzida manualmente pelo próprio, ao se registar na aplicação. Essa informação, para além de ser mais extensa que aquela introduzida aquando do registo através da rede social Facebook, deverá poder ser devidamente alterada posteriormente à finalização do registo e confirmação do email. Pelo que é necessário proceder à atualização da estrutura e comportamentos da página das definições de utilizador, dependendo do tipo de conta autenticada.

Todos os elementos previamente criados nos ecrãs de registo, foram adicionados ao ecrã das definições de utilizador (imagem, nome, género, data de nascimento, país, região, cidade e password).

No que toca à password, implementou-se um mecanismo de alteração da mesma. Caso o utilizador pretenda alterar a password, este terá de introduzir a password atual, seguida da nova password e a repetição da mesma. Depois de validados os campos da nova password, é efetuada uma comparação entre o *access token* gerado com base na password atual introduzida, com o *access token* presente na variável de sessão. Caso a password seja alterada com sucesso, a variável de sessão referente ao *access token* é então atualizada.

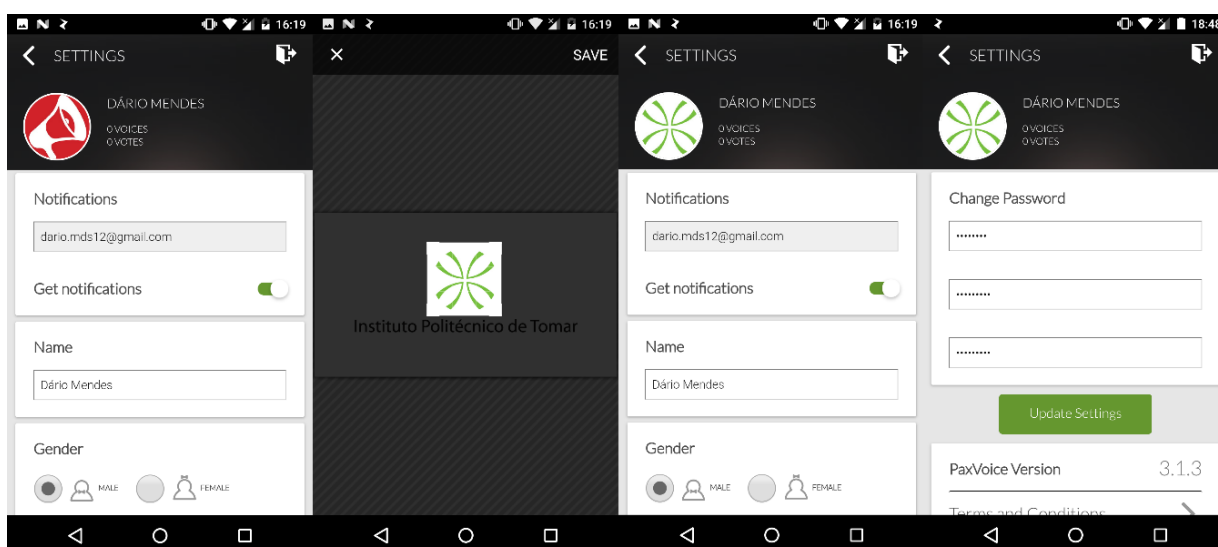


Figura 40 - Ecrã Definições de utilizador.

O maior desafio neste serviço, terá sido a implementação da ação do botão “*Update Settings*”. Pois o número de pedidos realizados ao servidor era variável. Isto porque eram dependentes das informações a serem atualizadas. E também, devido ao facto da informação não se encontrar dividida por fases, tal como no registo, partilhando o mesmo controlador (*settingsCtrl*).

De maneira a garantir que as informações que se encontram neste ecrã estão devidamente atualizadas, recorreu-se novamente à função AngularJS: *scope watch*.

4.3 Dificuldades

Surgiram algumas dificuldades em relação ao desenvolvimento dos projetos. Sendo que o principal contratempo, na maior parte dos casos, é a incompatibilidade entre plataformas para as soluções encontradas. Pois o trabalho desenvolvido, e testado em *browser*, nem sempre pode ser aplicado num dispositivo móvel. Devido a este facto, é sempre necessário realizar imensos testes em todas as plataformas, e no maior número de versões sistemas operativos possíveis. Revelando, deste modo, um dos únicos pontos fracos do desenvolvimento com base nas *frameworks* Cordova e Ionic.

A maior dificuldade ao nível de desenvolvimento encontrou-se no projeto Sou Cidadão. Nomeadamente a implementação do sistema de notificações. Passando por um longo e complexo processo de aprendizagem. Desde o estudo de várias versões de *plugins* (*phonegap-plugin-push* e *PushPlugin*), bem como a análise dos serviços de *push* para as três plataformas (GCM, APNS, MPNS). Contando ainda com respetivo o tratamento da informação referente às notificações por parte dos WebService em PHP. E ainda a sua integração com a plataforma Drupal.

5 Conclusões

O trabalho de estágio efetuado teve como base o desenvolvimento de serviços para projetos de aplicações móveis. Sendo estes, o projeto Sou Cidadão e PaxVoice. Ambos com funcionalidades e objetivos diferentes. Diferindo também, um pouco, nas tecnologias utilizadas para o seu desenvolvimento. Sendo que a aplicação Sou Cidadão foi baseada na *framework* Cordova, servindo como elo de ligação entre o cidadão e a sua Junta de Freguesia. Já aplicação PaxVoice, terá sido desenvolvida com base em Ionic contando com a integração de AngularJS. E tendo como objetivo a partilha fácil de opiniões publicas entre os seus utilizadores. O desenvolvimento nestes projetos permitiu a aquisição de conhecimentos, no que toca ao desenvolvimento de aplicações híbridas.

Particularmente, o desenvolvimento efetuado no projeto PaxVoice terá sido o que me despertou mais interesse. Especialmente pelo seu desenvolvimento em Ionic, que demonstrou ser uma mais-valia para o desenvolvimento de aplicações híbridas. E em especial pela integração do AngularJS. Pois esta *framework* de JavaScript permite uma metodologia de programação algo inovadora. Fornecendo novas formas de gestão da informação e interagindo, tanto ao nível do código HTML, como ao nível do código JavaScript.

Concluindo, após a realização deste estágio curricular enquadrado no Mestrado em Produção de Conteúdos Digitais, decorrido ao longo de nove meses, foi possível obter experiencia profissional. Pois ao fazer parte de uma equipa em ambiente de trabalho empresarial, na Compta, consegui entender a realidade do mercado de trabalho. Um pouco diferente do trabalho académico até então realizado. Onde foram aplicados os conhecimentos adquiridos ao longo dos estudos anteriormente realizados. Mas que também permitiu adquirir novas competências. Pelo que terá sido uma experiencia bastante enriquecedora, servindo como preparação para o meu futuro pessoal e carreira profissional.

6 Bibliografia

- 1) Compta, (2017). “Sobre Nós” [Consult. 2017-10-12]. Disponível em <<http://www.compta.pt/sobre-nos/>>
- 2) Tagusvalley, (2017). “TAGUSVALLEY Apresentação” [Consult. 2017-10-13]. Disponível em <<http://tagusvalley.pt/pt/tagusvalley/apresentacao/>>
- 3) Williamson, K., (2015) “Learning AngularJS: A Guide to AngularJS Development” 2015, O'Reilly Media, Inc.
- 4) W3S, (2017). “Build Ultra-Modern Web Apps with Angular Material” [Consult. 2017-10-12]. Disponível em <<https://www.toptal.com/angular-js/ultra-modern-web-apps-angular-material/>>
- 5) Drotner, K., Schroder, K (2010) “Digital Content Creation: Perceptions, Practices, & Perspectives” 2010, Peter Lang
- 6) Jones, M., (2017). “Introduction to JavaScript” [Consult. 2017-10-15]. Disponível em <<http://www.howtocreate.co.uk/tutorials/javascript/introduction>>
- 7) Saleh, H., Holmes E., Bray T., Yusuf S. (2016) “Mobile Application Development: JavaScript Frameworks” 2016, Packt Publishing Ltd
- 8) Yusuf, S., (2016) “Ionic Framework By Example” 2016, Packt Publishing Ltd
- 9) W3S, (2017). “CSS3 Introduction” [Consult. 2017-10-17]. Disponível em <https://www.w3schools.com/css/css3_intro.asp/>
- 10) Thakur, A., (2014). “Build your first web service with PHP, JSON and MySQL” [Consult. 2017-09-26]. Disponível em <<https://trinitytuts.com/build-first-web-service-php/>>
- 11) Drupal, (2017). “System Requirements – Web Server” [Consult. 2017-10-21]. Disponível em <<https://www.drupal.org/docs/8/system-requirements/web-server/>>
- 12) W3S, (2017). “jQuery Introduction” [Consult. 2017-10-17]. Disponível em <https://www.w3schools.com/jquery/jquery_intro.asp/>
- 13) Wargo, J., (2016) “Apache Cordova 4 Programming” 2015, Addison-Wesley Professional

- 14) Developer Mozilla, (2017). “Introduction to HTML5” [Consult. 2017-10-18].
Disponível em <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/Introduction_to_HTML5/>