



IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

RELATÓRIO DE ESTÁGIO

Licenciatura em Engenharia Informática

Micael Jorge Pires Martins

setembro | 2015



IPG

Politécnico
da Guarda

Polytechnic
of Guarda

*Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda*

RELATÓRIO DE ESTÁGIO

Website para Jornal

MICAEL JORGE PIRES MARTINS

RELATÓRIO PARA A OBTENÇÃO DO GRAU DE LICENCIADO

Em Engenharia Informática

Setembro de 2015



WebSite para Jornal

Micael Jorge Pires Martins

Aluno de Engenharia Informática, 3º ano, nº1010395

Datas do Estagio Curricular: Início: 15 Junho

Fim: 25 Setembro

Trabalho realizado no âmbito da disciplina Projeto de Informática do 2º semestre do 3º ano, da Licenciatura em Engenharia Informática do Instituto Politécnico da Guarda/ Escola Superior de Tecnologia e Gestão, orientado pelo Professor José Quitério

Instituto Politécnico da Guarda

Escola Superior de Tecnologia e Gestão

AV. Dr Francisco Sá Carneiro 50, 6300-559 Guarda

Contato: 271220120/ 220165

E-mail: estg-geral@ipg.pt

Dom Digital

Internet Services

Av. Rainha D. Amélia, 142 cave

6300-749 Guarda Portugal

Contato: +351 271 224 509

E-mail: domdigital@domdigital.pt



Plano de Estágio Curricular

O estágio curricular teve a duração de 12 semanas (3 meses), foi iniciado a 15 de junho de 2015 e finalizado a 25 de dezembro de 2015.

O estágio foi realizado na Instituição Dom Digital assim como o projeto desenvolvido, tendo como orientador o Professor José Quitério e ajudas por parte dos colaboradores da referida empresa.

Objetivo Inicial:

1. Contato com os responsáveis da instituição e conhecimento do projeto a desenvolver.
2. Recolha de informações importantes para a aplicação a desenvolver. (1 semana)
3. Formação na área de html5/css. (1 semana)
4. Formação em JavaScript. (1 semana)

Resumo

Inicialmente, para o desenvolvimento do projeto foi realizado um estudo prévio sobre os seus objetivos e enquadramento, mais tarde foi realizada formação nas linguagens de programação a utilizar de forma a desenvolver o trabalho proposto na empresa de forma organizada e construtiva.

No projeto foi desenvolvido um Web site para um jornal de notícias utilizando uma framework já existente utilizada na empresa.

A web site desenvolvido é responsive, ou seja adapta-se a qualquer resolução de ecrã, tornando-o dinâmico e adaptável para versão smartphone, Tablet e computador.

Na construção do Web Site temos a parte front-end e back-end o front-end inclui o html5, css, Java Script para criar a parte do interface com o utilizador ou Design da aplicação o back-end é a ligação aos servidores e à API ardina.api.

Utilizou-se html5/css para modificar a estrutura da referida framework através de uma matriz de linhas por colunas, adaptando-a assim para um jornal de notícias.

Desenvolveu-se o site em backbone que tem como base o JavaScript como linguagem de programação e deste modo a ligação ao ardina.api.

Por fim, para carregar o Web Site com conteúdo utilizou-se o ardina press, que pertence a família ardina criado na Dom Digital, e que tem como base o salesforce que é um software de gestão de relacionamento com clientes.

É de referir que o principal objetivo desta aplicação Web é o acesso à informação, o projeto foi desenvolvido com o intuito de fornecer uma fácil e rápida navegação com páginas interativas, e deste modo proporcionar aos utilizadores a informação pretendida.



Prefácio

Olhando para a evolução do mundo, podemos constatar que a informática é cada vez mais a base para tudo. Portanto, a sua evolução é diária e está presente no dia-a-dia. Isto para dizer que há cada vez mais linguagens de programação e portanto pode-se construir uma aplicação de diversas linguagens e formas diferentes.

Atualmente, os Web Sites presentes na web utilizam a tecnologia streaming, que é uma forma de distribuição de dados (Multimédia) através da Internet.

Neste trabalho, em sintonia com a empresa, apresenta-se uma forma de construir um site para um jornal de notícias construído em backbone. O backbone tem como base o JavaScript, e a nível estrutural é construído através de html5 e estilos css.

O conteúdo, neste caso as notícias é injetado na ardina.Press. A ardina.press foi desenvolvida pela empresa e é carregado no site através de chamadas á api.

Temos de dar especial atenção, em tornar o site multifuncional, para que este possa ser visualizado em diferentes resoluções de ecrã e simples de utilizar, organizado e interativo para que os utilizadores possam usufruir dele sem grandes dúvidas.



Agradecimentos

Entre muitas pessoas que me ajudaram ao longo da licenciatura destacam-se as seguintes: Pelo apoio e colaboração para a realização deste projeto expresso os sinceros agradecimentos ao professor orientador José Quitério e a todos os colaboradores da Dom Digital.

A nível pessoal agradeço aos meus pais, irmão, família e amigos que de uma forma ou de outra estiveram sempre presentes nesta etapa.

Índice Geral

1	Introdução.....	1
1.1	Caracterização da Instituição	2
1.1.1	Visão.....	2
1.1.2	Missão.....	2
1.1.3	Parcerias.....	3
1.1.4	Protocolos	4
1.1.5	Alguns marcos da Dom Digital	4
1.2	Cronograma:	6
1.3	Contextualização.....	6
1.4	Objetivo do projeto em contexto de Estágio.....	6
1.5	Organização do Relatório	7
2	Estado da arte.....	8
2.1	Soluções existentes no mercado.....	11
3	Metodologia e ferramentas utilizadas.....	14
3.1	Metodologia	14
3.2	Ferramentas e tecnologias utilizadas	15
3.2.1	Arquitetura.....	16
4	Implementação e trabalho desenvolvido	17
4.1	Plano de formação	17
4.2	Exemplo prático	17
4.3	Implementação do projeto	19
4.3.1	Introdução de dados no ardina.press.....	19
4.3.2	Estruturação e criação de páginas.....	22
4.3.3	Montagem do site (backbone) com chamadas ao ardina.....	31



4.4	Controlo de versões	39
5	Testes e Conclusão	40
5.1	Resultados Obtidos	40
5.2	Testes e validações.....	40
5.3	Conclusão.....	41
5.4	Implementações futuras	41
	Referências bibliográficas	42

Índice de Figuras

Figura 1 Salesforce parceiros tecnológicos	3
Figura 2-Google apps	3
Figura 3-Amazon web services	3
Figura 4- Metodologia Scrum (16)	14
Figura 5- Arquitetura	16
Figura 6- Ardina news	17
Figura 7- Exemplo prático	18
Figura 8- Ardina press view	19
Figura 9- Ardina press contents	20
Figura 10- Ardina press contents	20
Figura 11- Ardina press new content	21
Figura 12- Página header.html menu tags	22
Figura 13- Página principal resolução desktop	22
Figura 14- View esquerdacontent-noticia destaque	23
Figura 15- View centrocontent	23
Figura 16- Homeview	24
Figura 17- View direitacontent-publicidade	25
Figura 18- View guarda-seção de notícias	25
Figura 19- Post.html	26
Figura 20- Notícia principal da página post.html	27
Figura 21- Noticias relacionadas da página post.html	28
Figura 22- Mais notícias da página post.html	28
Figura 23- Noticias agrupadas por categoria na página intermédia.html	29
Figura 24- Seção dos vídeos na página visualizar.html	29
Figura 25- Página principal do website	31
Figura 26- Criação de coleção de dados para as tags	32
Figura 27- Inicialização da coleção de dados	33
Figura 28- Construção da view header	33
Figura 29- Função listMenu	34
Figura 30- Função render	34



Figura 31- Criação de rotas	35
Figura 32- Função home página inicial	35
Figura 33- Função home2- páginas intermédias	35
Figura 34- Função home3-videos	36
Figura 35- Funcionamento das rotas para tag desporto	36
Figura 36- View intermediatagsview	37
Figura 37- View homepostview	38
Figura 38- Rotas possíveis	38
Figura 39- SourceTree controlo de versões	39
Figura 40- Teste do website	40



Índice de tabelas

Tabela 1- Suporte a plataformas	12
Tabela 2- Disponibilidade de operações de suporte	13



Glossário

IPG: Instituto Politécnico da Guarda

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

WAP: Wireless Application Protocol

API: Application Programming Interface

CRM: Customer Relationship Management

APPS: application

PAAS: Platform as a Service

JSON: JavaScript Object Notation

MVC: Model view collection

1 Introdução

O objetivo deste trabalho consiste na elaboração de uma aplicação para a Internet de um jornal de notícias no distrito da Guarda.

Com o exponencial crescimento de utilização de serviços de internet assim como o uso de equipamentos para a aceder à informação de uma forma quase direta, então por estes motivos entre outros sentiu-se a necessidade de realizar este projeto. Uma outra razão tomada em conta foi o facto dos jornais web na região estarem um pouco ultrapassados, pois com a evolução diária do ramo da Informática surgem outras formas de construção da aplicação tornando-a mais rápida, leve e flexível.

Este jornal online vai conter os destaques, que são as últimas notícias injetadas no ardina press, e estas vão aparecer no canto superior esquerdo da página principal. Do lado direito vão aparecer notícias relacionadas com a referida anteriormente.

Em baixo as notícias dividem-se por categorias, Guarda, Desporto, Economia, Política, Cultura e Região contendo os vídeos do lado direito.

Quando o utilizador carregar numa dada notícia, essa mesma vai aparecer em grande com o respetivo texto em baixo, do lado esquerdo vão aparecer todas as notícias que sejam relacionadas com a referida e por baixo aparecem mais notícias podendo ser elas de outras categorias.

Este site contem ainda umas tags ou separadores com os nomes das categorias que torna o acesso às notícias correspondentes a estas mais fáceis aparecendo as notícias em causa da mais recente para a mais desatualizada.

1.1 Caracterização da Instituição

A Dom Digital – Novas tecnologias de informação LDA, é uma empresa que presta serviços com base na infraestrutura da Internet. Foi fundada em janeiro de 1997 na cidade da Guarda, Portugal. Enfoca a sua atividade no mercado empresarial criando soluções com resultados em Portugal, Espanha, Palop e Brasil.

1.1.1 Visão

Acreditam no meio Internet como potenciador da competitividade das empresas.

Acreditam num futuro onde as tecnologias da Internet farão cada vez mais parte da vida das empresas.

Entendem a Internet como um fenómeno global de grande dimensão, com múltiplas tecnologias e modelos de negócio.

Entendem a Internet – e a tecnologia em geral – como uma importante alavanca para o desenvolvimento da sociedade em geral.

1.1.2 Missão

Criar soluções criativas, com base em tecnologia fiável, que crie valor acrescentado ao negócio dos Clientes.

Prestar serviços de excelência aos Clientes, procurando atingir satisfação total.

Contribuir para o desenvolvimento da Internet portuguesa na sua componente social.

1.1.3 Parcerias

A empresa Dom Digital foi dos primeiros parceiros de Salesforce em Portugal (desde 2003) e desde então Salesforce é o principal parceiro tecnológico. Parceiros em consultoria (Salesforce Cloud Alliance Partner) e em desenvolvimento de aplicações na cloud (Appexchange Program Partner) como mostra a fig1 onde se destacam as aplicações por nós desenvolvidas: Dom Hotel, Speedy Data, Sol Global e Deal More.



Figura 1 Salesforce parceiros tecnológicos

Na Dom digital desenvolvem-se aplicações mobile para dispositivos Android e são revendedores Autorizados Google em Portugal (Google Apps Authorized Reseller) como mostra a fig2.



Figura 2-Google apps

O datacenter da empresa usa a infra-estrutura do líder mundial em IaaS - Amazon Web Services representado na fig3 permitindo vantagens para o cliente como, maior escalabilidade, melhor performance e mais redundância.



Figura 3-A Amazon web services

1.1.4 Protocolos

A Dom Digital tem vindo a fortalecer a sua parceria com um conjunto de instituições do ensino superior com o intuito de criar um maior interligação entre os universos empresarial e científico. Com estes protocolos a Dom Digital e as instituições de ensino superior comprometem-se a trabalhar em conjunto em projetos de investigação e desenvolvimento proporcionando formação avançada durante os estágios.

1.1.5 Alguns marcos da Dom Digital

De 1995 a 2000:

Foi criado o primeiro projeto comercial: elaboração do site do jornal Terras da Beira

Criado o primeiro serviço português de Auditoria a websites

Foi formalizada a constituição da empresa, Coordenação do "Plano Estratégico para a Sociedade da Informação para o Distrito da Guarda"

A Dom Digital foi finalista do concurso "Masters of Web" da Microsoft e lançou produtos da família ardina.com

De 2000 a 2005:

Foi criado o primeiro jornal português em WAP, Avaliação de 115 sites de jornais portugueses, ardina.web com versão para cidadãos com necessidades especiais.

A dom Digital tornou-se representante nacional de salesforce.com e obteve as suas próprias instalações.



De 2005 a 2010:

Lançamento do ardina.referral e ardina.payments, Primeiro Cliente dos PALOP e lançamento do SWS – SMS via web.

Lançamento de um novo serviço de eMarketing, Google Seminar Leader – e com isto a Dom Digital é parceira da Google na área da formação Google AdWords

De 2010 a 2013:

A Dom Digital implementou mobile web site do Grupo RTP E mobile web site com reservas mobile dos hotéis Vila Galé

Desenvolveu aplicação android do Grupo RTP e lança as aplicações DomHotel e SpeedyData, dentro da plataforma Salesforce (1).

1.2 Cronograma:

- 15 Junho a 25 Setembro 2015 (15 semanas)
Até 31 de julho
 - Tarefa 0: Conceitos em HTML e CSS: 1 semana
 - Tarefa 1: Conceitos em Javascript/BackBone.js: 3 semanas
 - Tarefa 2: Começar a desenvolver o website, apresentar uma 1 versão do site.

- Até 25 de Agosto
 - Tarefa 3: Concluir o desenvolvimento apresentar site final com Testes e validação
 -

1.3 Contextualização

Atualmente a generalidade dos jornais está presente na web através de tecnologia streaming. Streaming é uma forma de distribuição de dados (Multimédia) através da Internet. Para esta transmissão é necessário enviar o conteúdo para um servidor que irá realizar a codificação apropriada e a transmissão aos utilizadores.

A principal vantagem de um jornal online é o custo inferior ao de criação de um jornal tradicional, não tem limitações geográficas nem temporais. Com isto decidiu-se realizar uma aplicação para um jornal online possibilitando assim a transmissão de dados e multimédia via Internet em tempo real.

1.4 Objetivo do projeto em contexto de Estagio

De acordo com os responsáveis da Dom Digital e com a Unidade Curricular projeto, decidiu-se desenvolver um website responsivo para um jornal de notícias.

Foram definidas algumas etapas:

- Desenvolver a parte front-end e back-end de uma forma Responsive
- Frontend (html5, css, Java Script) para criar a parte do interface ou Design da aplicação.
- Back-end é a ligação à bases de dados, aos servidores e á API (ardina.api).

O objetivo final é obter um website, devidamente testado de modo a ser comercializada.

1.5 Organização do Relatório

Este relatório está estruturado em 5 capítulos. No primeiro capítulo é feita uma introdução e definição dos objetivos ao tema do projeto em contexto de estágio. É também descrita a contextualização do mesmo a organização do relatório e a caracterização da empresa onde foi desenvolvido o projeto. No segundo capítulo apresenta-se o estado de arte assim como soluções existentes no mercado. No terceiro capítulo apresenta-se a metodologia, e a descrição das ferramentas utilizadas. No quarto capítulo explica-se detalhadamente todas as fases de desenvolvimento do website. No quinto capítulo apresentam-se os resultados obtidos, os testes e as validações assim como a conclusão. Terminando com as sugestões de implementações futuras e referências bibliográficas.

2 Estado da arte

Inicialmente optou-se por realizar um plano geral do projeto, fazendo um estudo prévio das ferramentas a utilizar, e encontrar soluções existentes às mesmas

Plataforma Salesforce:

A plataforma em nuvem, torna o processo de fornecer funcionalidades para aplicações mais interessante. As aplicações que funcionam na nuvem afiguram-se cada vez menos como sites e mais como plataformas, o que significa que estas estão a começar a usar Interfaces de programação de aplicações (APIs).

Salesforce é um software de gestão de relacionamento com clientes (CRM) mas para além disso a salesforce redefine o sucesso no CRM, desde a automação da força de vendas para o marketing até o atendimento a clientes sendo assim uma das plataformas mais utilizadas nas áreas das vendas e pós vendas.

O salesforce permite às empresas procurar potenciais clientes, gerir todo o processo de venda e permite um acompanhamento constante dos seus clientes. Uma empresa pode registar as informações respetivas a um potencial cliente desde o primeiro contato até todo seu futuro relacionamento com a empresa. A plataforma possui ferramentas que possibilitam classificar o potencial cliente com base em critérios personalizados, programar reuniões e enviar e-mails sem sair da plataforma. Importante também pela questão da organização, onde todo material de um departamento, apresentações, documentos vai ficar acessível para toda a equipa de venda.

O Salesforce pode ser utilizado em qualquer lugar e em qualquer dispositivo. Resumidamente a disponibilidade e facilidade de acesso são vantagens extremamente importante nos dias de hoje (2).

Existem várias tecnologias salesforce entre as quais destacamos:

Sales cloud é uma aplicação de vendas (3).

Service Cloud é um serviço de apoio ao cliente, permite que os clientes tenham apoio total através de vídeos redes sociais, linha de apoio ao cliente em tempo real (4).

Marketing cloud é uma plataforma de monitorização de conversas, permitindo a uma empresa saber as opiniões trocadas nas redes sociais, blogs entre outros de um produto seu (5).

Community Cloud permite criar uma comunidade em que revendedores, distribuidores e parceiros possam partilhar informação (6).

Analytics Cloud permite recolher, analisar e distribuir informação crítica sobre um negócio (7).

Platform and Apps é uma ferramenta de criação de aplicações para qualquer sector de atividade na plataforma force.com para dispositivos móveis.

Force.com é a plataforma da salesforce.com para construir e executar aplicações empresariais na nuvem (8).

Mas tudo isto baseia-se no modelo PaaS que é um modelo comprovado para executar aplicações sem a necessidade de manutenção de hardware assim como infra-estrutura de software em qualquer empresa. Estas têm adotado soluções de PaaS genericamente o force.com devido a sua simplicidade, escalabilidade e confiabilidade (9).

Ardina.api: Faz parte da família ardina.com é produzida com base em force.com, que é a plataforma líder de desenvolvimento de aplicações empresariais na cloud.

Também é considerado um modelo de negócio/tecnologia pois promove a inovação em qualquer empresa, encontrando novas formas de monetizar conteúdos com as seguintes vantagens:

- Criar um ecossistema de parceiros para ajudarem a monetizar os conteúdos.
- Criar um ecossistema de developers para criar apps que usem conteúdos.
- Nova ferramenta para uma equipa de programadores mais rápida e com mais segurança.
- Total controlo de conteúdos, incluindo a definição das regras de utilização e os níveis de acesso (10).

Ardina.press: Faz parte da família ardina.com é produzida com base force.com. Force.com é a plataforma líder de desenvolvimento de aplicações empresariais na cloud. Ardina.press destina-se a empresas dos sector dos media (imprensa, rádio, tv, new media) com as seguintes funcionalidades: Criar, editar, publicar e gerir os conteúdos jornalísticos de forma rápida e intuitiva. Possui também características próprias que são elas: Gestor de conteúdos na Cloud, Gestão de elementos multimédia, Catalogação de artigos, Criação de tabelas, Rede social privada e Gestão de perfis.

Vantagens para o jornalista e Editor: Fácil e intuitivo de usar, funciona na cloud, permite mobilidade e Nível estatístico (11).

Backbone.js: É uma framework Javascript que vai fornecer componentes para melhorar a estrutura de aplicação web.

A nível estrutural baseia-se no conceito **mvc**:

M: Significa “model” que é o objeto que representa os dados.

V: Significa “view” e como o próprio nome indica é responsável pela parte visível.

C: Significa collection que é a coleção ou modelo responsável por lidar com os dados.

Backbone.js depende de uma framework underscore.js que vai fornecer recursos à aplicação JavaScript como suporte a templates e a recursos de programação (12).

2.1 Soluções existentes no mercado

Em relação ao force.com existem soluções no mercado tais como:

CloudBees: É uma plataforma recente que tem como objetivo simplificar o processo de desenvolvimento e execução de aplicações, utilizando para isso o conceito de computação em nuvem. A computação em nuvem é uma nova técnica que está a ser muito usada devido ao facto de não se precisar mais de instalar ou armazenar conteúdo (programas e aplicações) no computador do utilizador ou num servidor. Esse conteúdo fica disponível na internet.

Cloud Foundry: é uma plataforma de hospedagem e um serviço PaaS cujos softwares suportam datacenters virtualizados.

Red Hat openShift: está entre os mais populares servidores de aplicações Java.

Apresenta-se a seguir duas tabelas onde se faz a comparação das referidas soluções a nível de suporte a plataformas (tabela1) e a nível de Disponibilidade de opções de suporte (tabela2).

	CloudBees	Cloud Foundry	OpenShift
Tomcat	Sim	Sim	Sim
Java SE	Sim	Sim	Sim
Java EE	Sim	Não	Sim
Suporte a bibliotecas Java padrão	Sim	Sim	Sim
Acesso ao sistema de arquivos	Sim	Sim	Sim
Acesso a threads	Sim	Sim	Sim
Conexão a rede externas	Sim	Sim	Sim
Mysql	Sim	Sim	Sim
Bancos de dados	Externo	Externo	Externo
Suporte a big Data	Externo	Externo	Externo
Deploy sem frameworks especiais	Sim	Não	Sim
Amigável para migração de aplicações existentes	Sim	Não	Sim
Portabilidade de aplicações	Alta	Moderada	Alta
Pronto para produção	Sim	Beta	Beta

Tabela 1- Suporte a plataformas (13)

	CloudBees	Cloud Foundry	OpenShift
Serviço básico gratuito	Sim	Não se aplica	Sim
Custo para aplicações web com baixo tráfego	Zero	Zero	Zero
Suporte a múltiplos ambientes na nuvem	Não	Planeado	Planeado
Nuvem privada	Beta	Sim	Planeado
Suporte	E-mail e telefone	Chamadas via web	Fórum
Qualidade de suporte	Boa	Boa	Boa

Tabela 2- Disponibilidade de operações de suporte (13)

Optamos por utilizar o force.com porque na empresa em questão é a utilizada visto que é certificada em salesforce.

Em relação ao backbone poderia ter-se usado simplesmente jquery que é uma outra biblioteca do JavaScript.

Utilizamos backbone pois a aplicação desenvolvida exige muito JavaScript e houve a necessidade de manter uma estrutura para ficar mais fácil de manipular o código.

Com o backbone a aplicação torna-se escalável por outras palavras diz-se que o código passa a ter uma estrutura e com isto a parte front-end fica melhorada para um nível mais elevado (14).

É de referir que neste trabalho para um controlo de versões usamos o Bitbucket existindo várias soluções tais como o Mercurial e o gitHub mas também se utilizou este porque é o software utilizado pela empresa (15).

3 Metodologia e ferramentas utilizadas

3.1 Metodologia

Usamos a metodologia scrum em conformidade com a empresa que faz parte da metodologia de desenvolvimento de software ágil como mostra a figura 4.



Figura 4- Metodologia Scrum (16)

No Scrum, os projetos são divididos em ciclos que normalmente são mensais mas também podem ser semanais chamados de Sprints. O Sprint representa um conjunto de atividades que devem ser executadas num tempo definido, neste caso mensal ou semanal.

3.2 Ferramentas e tecnologias utilizadas

Nesta secção vamos descrever o conjunto das seguintes tecnologias:

Force.com: plataforma de computação em nuvem escalável e segura pertencente ao salesforce baseada em paaS (plataforma como serviço). Com isto tem-se toda a infraestrutura necessária para desenvolver e executar a aplicação através da Internet. Ferramenta Poderosa pois é escalável, segura e permite desenvolver aplicações instantaneamente, sem qualquer infraestrutura externa (8).

Ardina.api: é um modelo de negócio e uma tecnologia que promove a inovação sendo muito utilizado para monetizar conteúdos e é composto por dois módulos:

Módulo de negócio: Para as empresas encontrarem os melhores modelos de negócio para rentabilizar ao máximo os seus conteúdos através da API

Módulo técnico: Projeta-se tecnicamente uma API, integrando-a com plataforma informática numa dada empresa ou com a família de produtos ardina.com

Neste projeto muito importante devido ao facto de fazer a ligação do conteúdo injetado na ardina.press para o site (10).

Ligação esta feita através de um url onde se vai buscar o conteúdo.

Ardina.press: Destina-se a empresas dos sectores dos Mídias (imprensa, rádio, tv, new media) com as seguintes funcionalidades: Criar, editar, publicar e gerir os conteúdos jornalísticos de forma rápida e intuitiva (11).

Neste trabalho o Ardina.press acaba por ser a base de dados que guarda os conteúdos jornalísticos injetados com as seguintes vantagens: Fácil e intuitivo de usar, toda a informação centralizada num único sítio, mobilidade e compatibilidade.

Sublime Text2: É um IDE desintegrado e código-fonte multiplataforma, escrito em linguagem C++. Usado no projeto para desenvolver código em backbone.

Xampp/ apache: Coleção de programas que permitem gerar um servidor remoto, gerando um porto específico, este é atribuído á ligação e desta forma verificam-se os resultados no browser.

Bitbucket: Sistema de controlo de versões distribuída. Bitbucket é muito importante no desenvolvimento do projeto tanto por motivos de segurança como para a colaboração entre equipa (15).

3.2.1 Arquitetura

Na fig5 apresenta-se o diagrama de instalação onde é possível verificar que para o website receber conteúdo têm que passar pela ardina.api pois esta serve como ponte entre a ardina.press e o website através das ligações efetuadas na mesma. Verifica-se também a ligação feita por parte dos utilizadores ao website e todo este processo é feito na nuvem/internet.

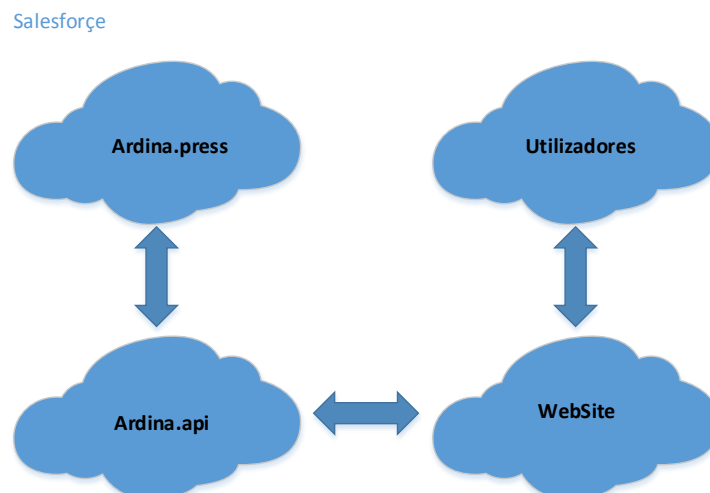


Figura 5- Arquitetura

4 Implementação e trabalho desenvolvido

4.1 Plano de formação

O início do estágio em conformidade com a empresa começou com um plano de formação que durou cerca de 3 semanas.

Foi realizada formação em html5/css, Javascript e backbone realizando uns cursos online para o referido.

4.2 Exemplo prático

Para por em prática os fundamentos aprendidos na formação começou-se por realizar um exemplo prático.

Este exemplo prático consistiu em adaptar uma estrutura em html5 (fig6) já construída para um jornal de notícias com dados vindos da ardina. (fig7).



Figura 6- Ardina news

Começou-se por analisar o código html e dividir tudo em views diferentes em que todas as matrizes representadas por quadrados ou retângulos são views diferentes.

De um modo geral as tags (Desporto, cultura, visualizar, sociedade, politica e guarda) constituem uma view. Em baixo a noticia representada com maior dimensão é outra view as 4 notícias ao lado são outra view e o mesmo para a secção dos vídeos.

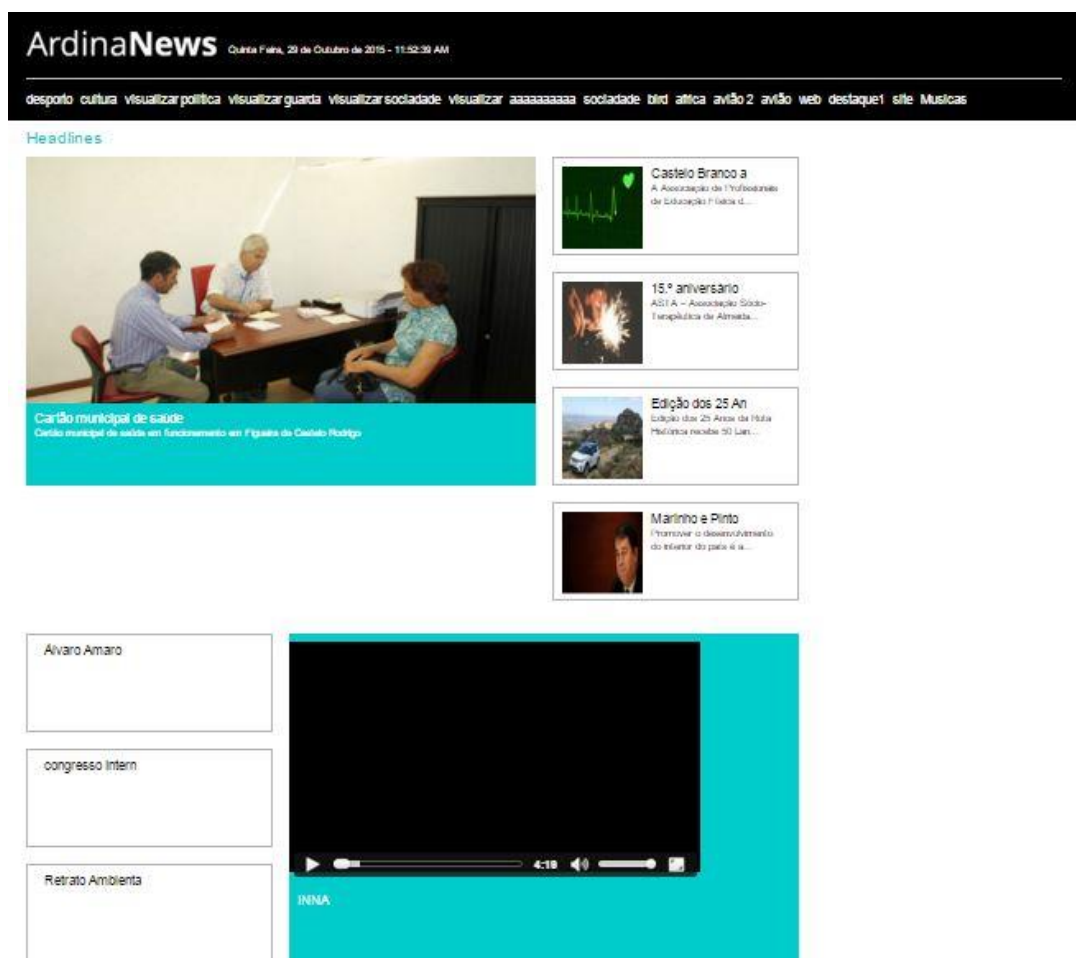


Figura 7- Exemplo prático

4.3 Implementação do projeto

4.3.1 Introdução de dados no ardina.press

Para ter dados reais foi necessário injetar conteúdo na ardina.press. Nesta introduziu-se notícias e vídeos.

O ardina.press está dividido por categorias, para uma melhor organização devido ao facto de quando carregarmos as notícias no website estas veem todas seguidas colocamos as notícias correspondentes a um determinado tema por ordem, o mesmo foi feito para os vídeos.

A fig8 é a página inicial do ardina.press onde estão as categorias.

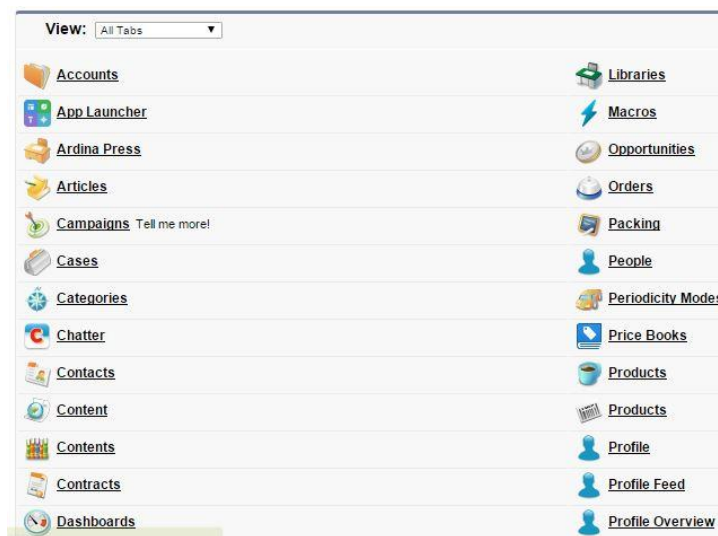


Figura 8- Ardina press view

Para introduzir os referidos dados entrou-se na categoria contents (conteúdo) que é o local onde se encontram todos os dados já inseridos, a seguir seleciona-se o botão go! Expresso na fig9.

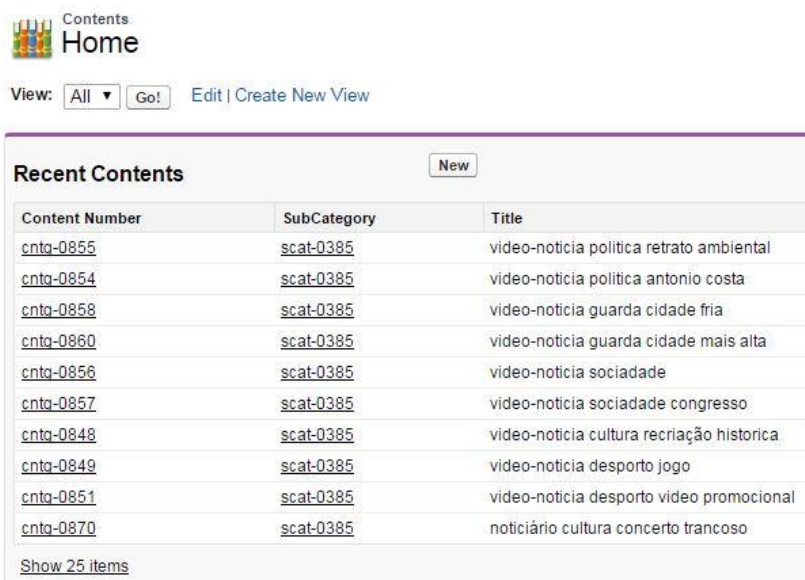


Figura 9- Ardina press contents

Para inserir um novo conteúdo a lista já existente seleciona-se New contente (fig10).

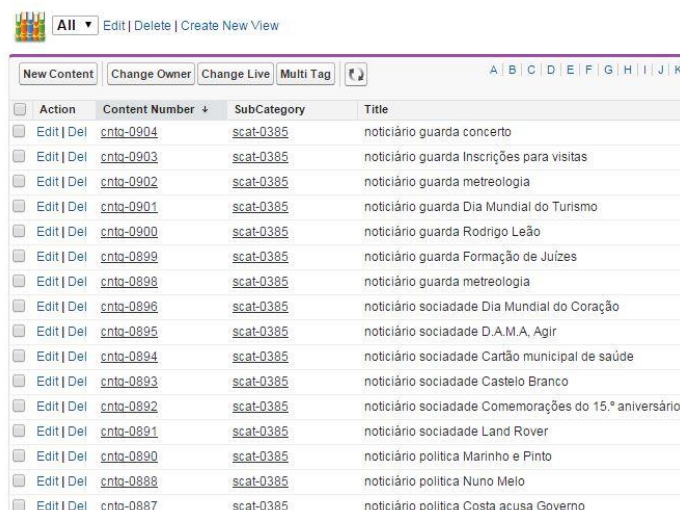
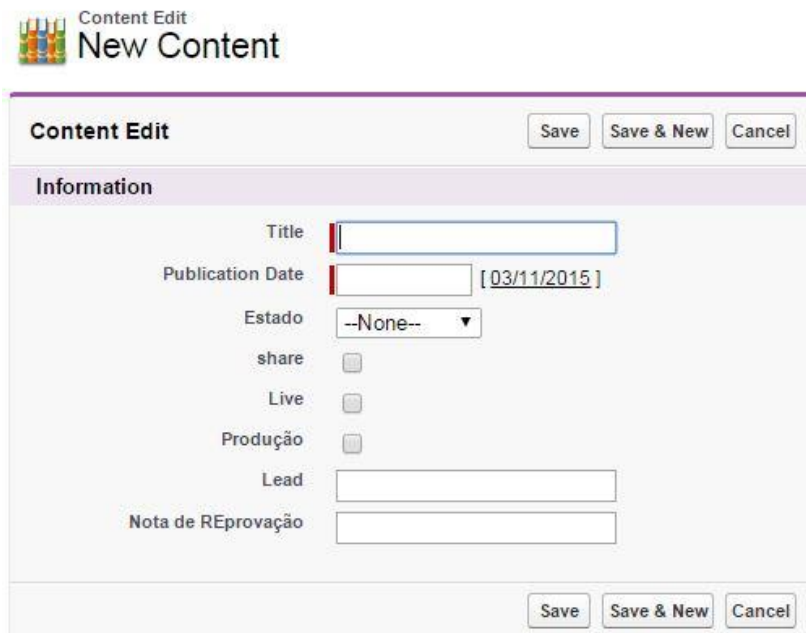


Figura 10- Ardina press contents

Por fim introduzem-se os dados correspondentes ao conteúdo que se pretende introduzir, título, data, texto é o nome que vai aparecer para identificar o conteúdo na coleção como mostra a fig11 procedendo depois de seguida ao carregamento das imagens ou vídeos sobre a mesma assim como o título, texto, legenda e autor.



Content Edit
New Content

Content Edit Save Save & New Cancel

Information

Title

Publication Date [03/11/2015]

Estado --None--

share

Live

Produção

Lead

Nota de REprovação

Save Save & New Cancel

Figura 11- Ardina press new content

4.3.2 Estruturação e criação de páginas

Em sintonia com a empresa e a partir do trabalho desenvolvido anteriormente na Figura 7 vamos construir o site final. Partindo do HTML já existente dividiu-se o mesmo em diferentes views criando-se algumas páginas diferentes:

Header.html: Está pagina acaba por ser um menu e está sempre presente durante toda a navegação no website. Contêm o título (ArdinaNews), as horas fornecidas pelo sistema usando uma função para o efeito e as tags: Desporto, Cultura, Visualizar, Sociedade, Politica e Guarda que são dados vindos da ardina.api. Esta página foi construída usando uma matriz de 16 colunas que é o tamanho total da página em largura na resolução desktop, como podemos ver na fig12.



Figura 12- Página header.html menu tags

Como se pode verificar na fig13 o website é responsivo, Quando a resolução de ecrã diminui a matriz de colunas da página header.html e contente.html vai ser modificada aparecendo um ícone que quando selecionado devolve a lista das tags. Isto possível através de estilos css e é usado quando a página é carregada na versão Smartphone (fig13).

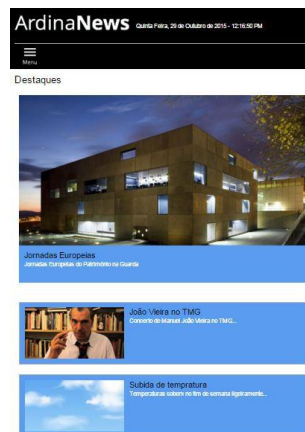


Figura 13- Página principal resolução desktop

Content.html: É a página principal a par do header.html. A página header.html é fixa e aparece durante toda a navegação, já a contente.html é desenhada quando a web site é carregado.

Em relação ao html a página contente.html está dividida em diferentes views: View Esquerdacontent- Aqui aparecem as notícias em destaque que são as últimas injetadas na ardina press e utilizam uma matriz de 8 colunas em largura (fig14).

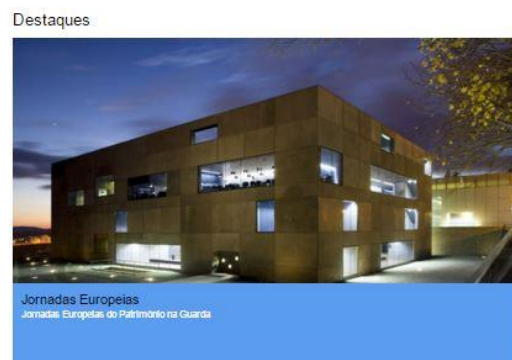


Figura 14- View esquerdacontent-noticia destaque

View Centrocontent- Utilizam uma matriz de 4 colunas em largura onde vão aparecer as notícias recentes representado na fig15.

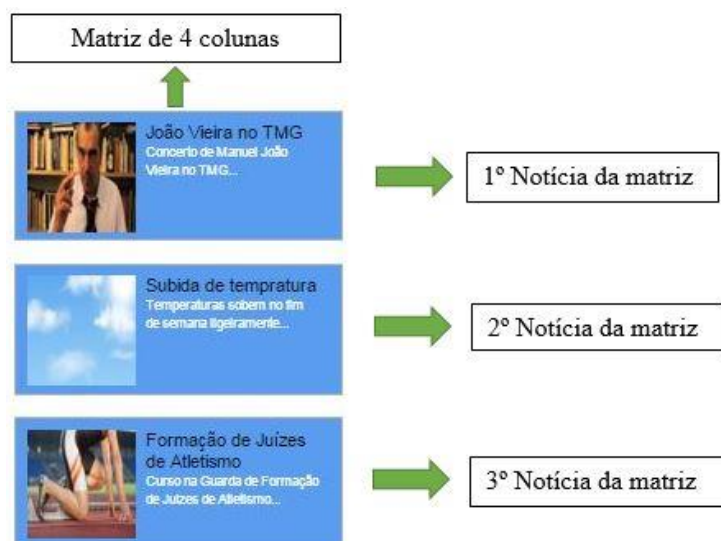


Figura 15- View centrocontent

Na view centrocontent aparecem as 3 notícias mais recentes. No código da fig16, podemos ver a HomeView que está associada por id a todas as views criadas nas páginas html, neste caso centrocontent (linha 568). Assim controlamos e manipulamos os dados vindos da ardina.api, mandando-os imprimir no sítio correto.

```

524 HomeView = Backbone.View.extend({
525   el: "#content",
526   template: "content.html",
527   template: contentTpl,
528   initialize: function(options) {
529     this.listenTo(this.collection, 'sync', this.listcont);
530   },
531   events: {
532     "click #menuicon": "menuClickFunc",
533     "click .clickContent": "showContent",
534     "click .articlehome": "selectNew2"
535   },
536   selectNew2: function(ev){
537     var self = this;
538     console.log(ev.currentTarget.attributes[0].value,"idNew;;;",idnew);
539     if(idnew != ev.currentTarget.attributes[0].value ){
540       idnew = ev.currentTarget.attributes[0].value;
541       this.homepostView = new HomepostView({id: ev.currentTarget.attributes[0].value});
542       this.homepostView.render2(ev.currentTarget.attributes[0].value);
543       console.log(ev.currentTarget.attributes[0].value);
544       idnew = ev.currentTarget.attributes[0].value;
545     }
546   },
547   listcont: function(){
548     var self = this;
549     console.log("this.collection fgdghdfgh",this.collection);
550     var cont = 0;
551     for(var i = 0; i < self.collection.models[0].attributes.data.contents.length; i++){
552       var model = self.collection.models[0].attributes.data.contents[i].article;
553       var modelimages = self.collection.models[0].attributes.data.contents[i].files[0];
554       if(model != null && modelimages != null){
555         if(cont ==0){
556           var restricaotitulos = model.title.substring(0,100);
557           var restricaocaracter1 = model.lead.substring(0,80); //limita os caracteres a
558           self.$("#esquerdacontent").append('<article data-type='+model.id+' class="has
559         }
560         if( cont==0 && cont<3){
561           var image = 'img/100x100a.jpg';
562           if (modelimages != null){
563             //image = modelimages[0].url;
564             var restricaotitulos = model.title.substring(0,100);
565             var restricaocaracter1 = model.lead.substring(0,80);
566             self.$("#centrocontent").append(' <article data-type='+model.id+' class="has-
567           }
568         }
569       }
570     }

```

Figura 16- Homeview

Foi adicionada uma classe para desenhar 4 colunas do lado direito onde vai aparecer publicidade como mostra a fig17.

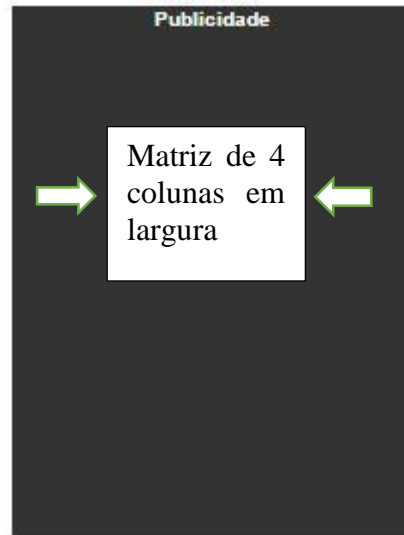


Figura 17- View direitacontent-publicidade

Mais abaixo podemos visualizar na fig18 a seção de notícias para Guarda estruturada com 3 views de quatro colunas cada e a seção de vídeos com 1 view de 4 colunas para o efeito.

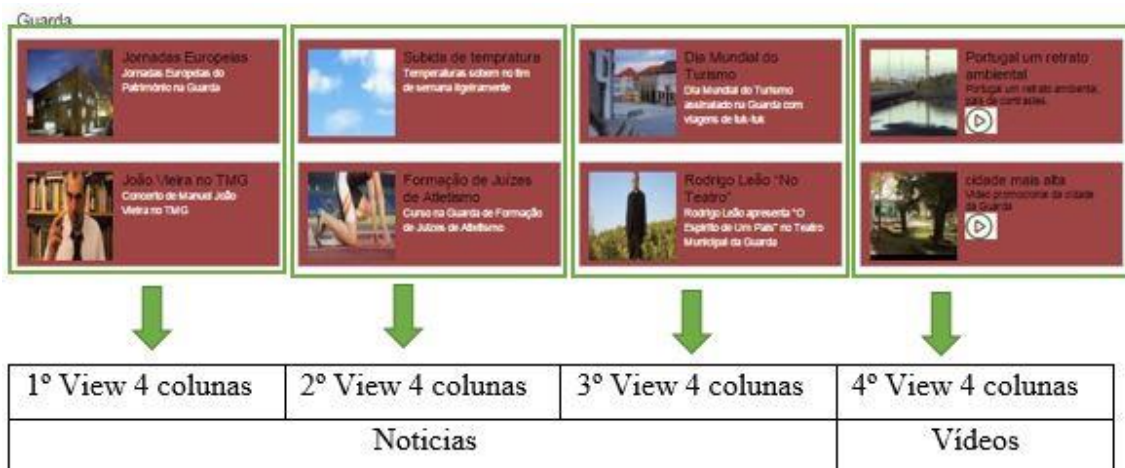


Figura 18- View guarda-seção de notícias

A estruturação da seção de notícias para Sociedade, Política, Desporto e cultura é precisamente a mesma do que a visualizada na fig18, mudando apenas de linha (*row*) para mudar de categoria.

post.html: É a página chamada quando o utilizador do website carrega numa notícia aparecendo depois a mesma como notícia principal da página post (fig19).

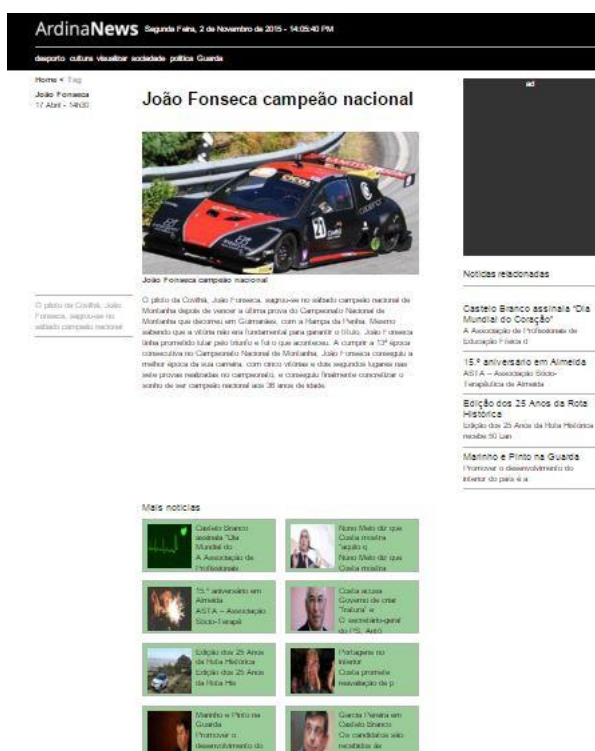


Figura 19- Post.html

Nesta página tem-se:

Notícia principal post representada na fig20 e contém a seguinte estrutura:

1º View que ocupa 3 colunas onde vai aparecer o autor.

2º View com 8 colunas onde aparece o título, imagem e legenda correspondente a notícia em questão.

3º View com 8 colunas onde aparece o texto correspondente a notícia em questão



The image shows a screenshot of a news article layout with three annotated views:

- 1º view:** A box on the left containing the breadcrumb "Home < Tag", the author "João Fonseca", and the date/time "17 Abril - 14h30".
- 2º view:** A box on the left with a green arrow pointing to the main article content. The content includes the title "João Fonseca campeão nacional", a photograph of a red and black rally car, and a small caption "João Fonseca campeão nacional".
- 3º view:** A box on the left with a green arrow pointing to the main text of the article. The text begins with "O piloto da Covilhã, João Fonseca, sagrou-se no sábado campeão nacional de Montanha depois de vencer a última prova do Campeonato Nacional de Montanha que decorreu em Guimarães, com a Rampa da Penha. Mesmo sabendo que a vitória não era fundamental para garantir o título, João Fonseca tinha prometido lutar pelo triunfo e foi o que aconteceu. A cumprir a 13ª época consecutiva no Campeonato Nacional de Montanha, João Fonseca conseguiu a melhor época da sua carreira, com cinco vitórias e dois segundos lugares nas sete provas realizadas no campeonato, e conseguiu finalmente concretizar o sonho de ser campeão nacional aos 38 anos de idade."

Figura 20- Notícia principal da página post.html

Noticias Relacionadas post representada na fig21 e têm a seguinte estrutura:

1º View 4 colunas para a publicidade

2º View 4 colunas para as notícias relacionadas

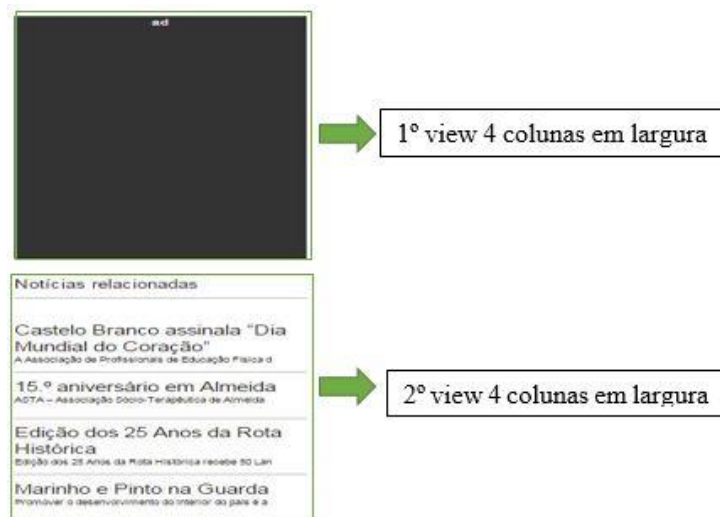


Figura 21- Noticias relacionadas da página post.html

Mais Noticias post representado na fig22 têm a seguinte estrutura:

1º View 4 colunas para a coluna da esquerda

2º View 4 colunas para a coluna da direita



Figura 22- Mais notícias da página post.html

Intermedia.html: é a página que aparece quando se carrega numa tag/categoria do menu da página header.html com a exceção do visualizar, (Desporto, Cultura, Sociedade, Política e Guarda) são as referidas tags. Quando uma destas é selecionada é desenhada uma outra página com as notícias referentes à tag ou categoria representado na fig23.



Figura 23- Notícias agrupadas por categoria na página intermédia.html

Visualizar.html: é a página representada na fig24 que aparece quando selecionada a tag/categoria visualizar do menu da página header.html ou quando se seleciona um vídeo relacionado com uma noticia na pagina principal.

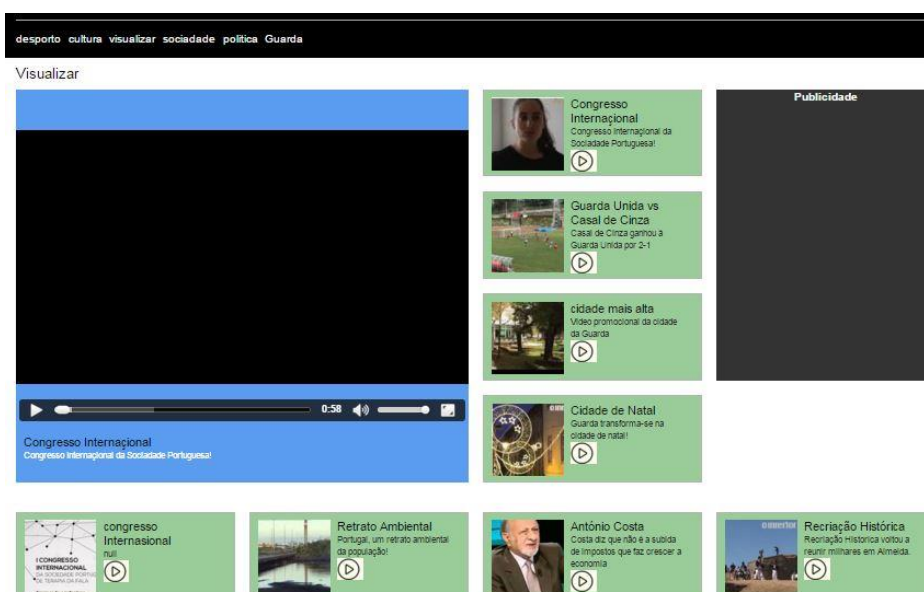


Figura 24- Seção dos vídeos na página visualizar.html

Como referido acima então quando um vídeo é selecionado é desenhada uma página com o vídeo correspondente em destaque que ocupa 4 colunas mais á esquerda e com todos os outros vídeos referentes a todas as notícias existentes na ardina.press podendo ser visualizada na fig24.

É de referir que no backbone não existem hiperligações. Construíram-se as páginas.html com o intuito de dar a estrutura ao website pois é dentro destas páginas que se vão imprimir os dados vindos da ardina.press numa view específica.

A navegação no website é definida no backbone através de rotas, pois estas têm eventos associados em que de uma forma resumida nestes eventos mandamos desenhar ou apagar as páginas.html como é possível visualizar a seguir.

4.3.3 Montagem do site (backbone) com chamadas ao ardina

Para uma melhor compreensão na fig25 está representado o website quando é carregado, representado também o menu das tags/categorias exemplo explicado a seguir.

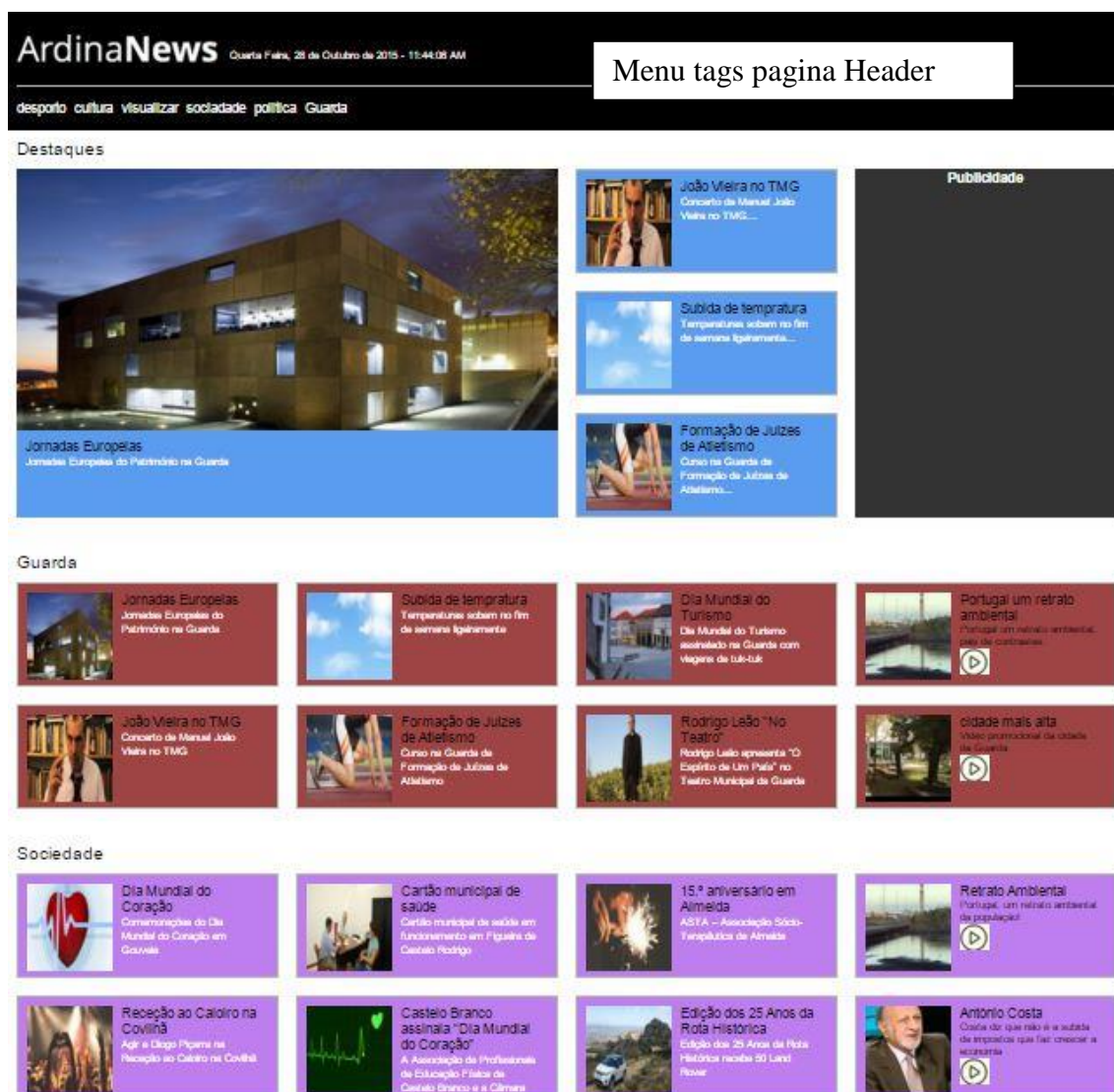


Figura 25- Página principal do website

O `app.js` é onde o código para a construção do site se encontra. Em seguida optamos por dar um exemplo concreto dos passos efetuados no backbone para a construção do menu das tags/categorias.

Passo 1: Para se aceder aos dados vindos do ardina foi necessário criar a coleção de dados representados na fig26. Nesta coleção existe um `url` responsável pela mesma. É de salientar que se usa JSON como sintaxe para armazenamento e troca de dados.

Existe ainda o `initialize` que é uma função onde são inicializadas as coleções de dados e o `sync` que é a sincronização através dos tipos de dados que me devolve o conteúdo vindo da `ardina.press` neste caso as tags (Desporto, Cultura, Visualizar, Sociedade, Política e Guarda). Se correr tudo bem no acesso ao conteúdo é devolvida uma mensagem de sucesso caso contrário uma mensagem de erro

```
var TagsCollection = Backbone.Collection.extend({
  initialize: function (options){
    /*this.type = options.type || 'todos';
    this.filters = options.filters || ''*/
  },
  sync: function(method, collection, options){
    if(method == "read") {
      $.ajax({
        type: "GET",
        url: "https://ardina-api.herokuapp.com/api/1/tags?user_key="+myUserKey,
        crossDomain: true,
        dataType: 'json',
        success: function (data) {
          // console.log("---->tags",data);
          options.success(data, 'successtags', null);
        },
        error: function(data) { console.log("errortags::",data); }
      });
    }else{
      Backbone.sync(method, collection, options);
    }
  }
});
```

Figura 26- Criação de coleção de dados para as tags

Passo 2: Iniciar a coleção de dados representada na fig27

Aqui inicializa-se a view, definindo esta com o nome tagscollection e vai ser a coleção de dados vindos da ardina.api.

```
initialize: function() {  
  //para tags(header)  
  var tagsCollection = new TagsCollection();  
  this.headerView = new HeaderView({collection: tagsCollection});  
  this.headerView.render();  
  tagsCollection.fetch();  
}
```

Figura 27- Inicialização da coleção de dados

Passo 3: Construção da view header representado na fig28

HeaderView é a página que vai receber os dados da tagscollection criada anteriormente.

Na fig28 da linha 234 á linha 237 construiu-se uma view chamada de **headerview**. Esta contem uma página chamada de header e vai receber como template a página header.html já construída anteriormente.

Na linha 244 tem-se os eventos associados a esta página para quando ocorre o referido evento click. Na linha 249 á 261 está representada uma função que é chamada quando ocorre o evento click da linha 245. Este faz com que o headerview seja responsivo alterando a estrutura matricial inicial dependendo da resolução de ecrã.

```
234 Headerview = Backbone.View.extend({  
235   el: "#header",  
236   templateFileName: "header.html",  
237   template: headerTpl,  
238  
239   initialize: function() {  
240  
241     this.listenTo(this.collection, "sync", this.listMenu);  
242   },  
243  
244   events: {  
245     "click #menuicon" : "menuClickFunc",  
246     "click .clickMenu" : "showPageMenu"  
247   },  
248   //para quando utilizo a resolucao do telemovel aparecer  
249   menuClickFunc: function(){  
250     console.log("menu");  
251     if($("#menuicon").hasClass("active")){  
252       $("#menuicon").removeClass("active");  
253     }else{  
254       $("#menuicon").addClass("active");  
255     }  
256  
257     $("#mainmenu ul").toggleClass("inline");  
258     $("#mainmenu ul").toggleClass("flyover");  
259     $("#mainmenu").toggle();  
260  
261   },  
262 });
```

Figura 28- Construção da view header

A fig29 mostra como se imprimem os dados vindos da ardina na respetiva view criada para o efeito. Destaca-se da linha 270 á linha 273 como o caminho percorrido para a transição dos dados da ardina para o website.

As tags são percorridas através do ciclo for e só são devolvidas as pretendidas. Estas são impressas na estrutura criada anteriormente neste caso na página header.html e na view que contém o id putMenus.

O model é o modelo de dados vindo da ardina assim como o model.name que representa o nome dos dados vindos da ardina.

```
264 listMenu: function(){
265     var self = this;
266     console.log("this.collection",this.collection);
267
268     var arraytags = ["Guarda", "desporto", "cultura", "politica", "sociadade", "visualizar"]
269
270     for(var i = 0; i < self.collection.models[0].attributes.data.tags.length; i++){
271         for(var y = 0; y < arraytags.length; y++){
272             if(arraytags[y]==self.collection.models[0].attributes.data.tags[i].name){
273                 var model = self.collection.models[0].attributes.data.tags[i];
274                 self.$("#putMenus").append('<li class="clickMenu" data-type="'+model.name+'" id="'+model.id+'><a href="#">'+model.name+'</a></li>');
275             }
276         }
277     }
278 }
```

Figura 29- Função listMenu

Na fig30 é apresentada a função render que tem como objetivo desenhar a página com todos os parâmetros construídos nos passos anteriores.

```
render: function() {
    $(this.el).html(_.template(this.template));
}
```

Figura 30- Função render

Passo 4: Em seguida são criadas as rotas (routes), visível na fig31, estas são as responsáveis pela navegação no website.

```
138 ▼ var ApplicationRouter = Backbone.Router.extend({
139     //sao criadas as rotas
140 ▼     routes: {
141         "home": "home",
142         'home/:id': 'home2',
143         'home1/:id': 'home3'
144     },
145 }
```

Figura 31- Criação de rotas

A partir da fig31 explicou-se os passos efetuados para a construção das rotas, visíveis nas próximas figuras. Home fig32: função que vai chamar e desenhar a view inicial (homeview).

```
179     home: function() {
180         /*if(id){
181             this.homeView = new HomeView({id:id});
182             this.homeView.render();
183         }else{
184             this.homeView = new HomeView();
185             this.homeView.render();
186         }*/
187         this.homeView = new HomeView();
188         this.homeView.render();
189     },
190 }
```

Figura 32- Função home página inicial

Home2: representada na fig33 é a função que vai chamar e desenhar a view noticiaprincipalview, noticiasrelacionadasview e maisnoticiasview. No website estas três views vão dar origem á Figura 19

```
192     home2: function(id) {
193         console.log('-----',this.contentCollection.models[0].attributes);
194         var x = this.contentCollection.models[0].attributes.data.contents.length;
195         for(var i = 3; i < x; i++){
196             console.log(this.contentCollection.models[0].attributes.data.contents[i].article.id,':::',id);
197             if(this.contentCollection.models[0].attributes.data.contents[i].article.id == id){
198                 //this.homeView = new HomeView({model: this.contentCollection.models[0].attributes.data.contents[i]});
199                 this.noticiaprincipalView.render();
200                 this.noticiaprincipalView.render2(this.contentCollection.models[0].attributes.data.contents[i]);
201             }
202             this.contentCollection = new ContentCollection();
203             this.noticiasrelacionadasView = new NoticiasrelacionadasView({collection: this.contentCollection});
204             this.noticiasrelacionadasView.render();
205             this.maisnoticiasView = new MaisnoticiasView({collection: this.contentCollection});
206             this.maisnoticiasView.render();
207             this.contentCollection.fetch();
208         }
209     },
210 },
211 }
```

Figura 33- Função home2- páginas intermédias

Home3 representada na fig34 é a função que vai chamar e desenhar a view visualizarview2 correspondente aos vídeos.

```
212     home3: function(id) {
213         console.log('----',this.contentvideocollection);
214         var x = this.contentvideocollection.models[0].attributes.data.files.length;
215         for(var i = 0; i < x; i++){
216             console.log(this.contentvideocollection.models[0].attributes.data.files[i].id,'::::::::::',id);
217             if(this.contentvideocollection.models[0].attributes.data.files[i].id == id){
218                 //this.homeView = new HomeView({model: this.contentCollection.models[0].attributes.data.contents[i]});
219
220                 console.log("entra");
221
222                 this.visualizarView2 = new VisualizarView2({collection: this.contentvideocollection});
223                 //sthis.visualizarView.render();
224
225                 this.visualizarView2.render2(this.contentvideocollection.models[0].attributes.data.files[i]);
226
227                 this.contentvideocollection.fetch();
228             }
229         }
230     }
231 });
232
```

Figura 34- Função home3-videos

No exemplo a seguir fig35 mostra-se melhor o funcionamento das rotas usando um exemplo concreto. Quando um dado utilizador clicar na tag/categoria desporto é inicializada a view intermediatagsview linha 474 e são filtradas todas as notícias relacionadas com a categoria desporto.

```
472     if(ev.currentTarget.attributes[1].value== "desporto"){
473         this.contagsCollection = new ContagsCollection({tag: ev.currentTarget.attributes[1].value});
474         this.intermediatagsView = new IntermediatagsView({collection: this.contagsCollection});
475         this.intermediatagsView.render();
476         this.contagsCollection.fetch();
477     }
```

Figura 35- Funcionamento das rotas para tag desporto

A seguir fig36 as notícias que estão associadas a tag desporto são impressas na página intermedia.html. Quando uma notícia é selecionada ocorre o evento selectinter, que vai enviar o id da notícia para a view homepostview.

```
848 IntermediatagsView = Backbone.View.extend({
849   el: "#content",
850   template: "intermedia.html",
851   template: intermediaTpl,
852   initialize: function(options) {
853     this.listenTo(this.collection, 'sync', this.listcont);
854   },
855
856   events:{
857     "click .clickContent" : "showContent",
858     "click .articleinter" : "selectinter"
859   },
860
861   selectinter: function(ev){
862     var self = this;
863     console.log(ev.currentTarget.attributes[0].value,"idNew;;;",idnew);
864     if(idnew != ev.currentTarget.attributes[0].value ){
865       idnew = ev.currentTarget.attributes[0].value;
866       this.homepostView = new HomepostView({id: ev.currentTarget.attributes[0].value});
867       this.homepostView.render2(ev.currentTarget.attributes[0].value);
868
869       console.log(ev.currentTarget.attributes[0].value);
870       idnew = ev.currentTarget.attributes[0].value;
871     }
872   },
873
874   listcont: function(){
875     var self = this;
876     console.log("33442345this.collection",this.collection);
877     var cont = 0;
878     for(var i = 0; i < self.collection.models[0].attributes.data.contents.length; i++){
879       var model = self.collection.models[0].attributes.data.contents[i].article;
880       var modelimages = self.collection.models[0].attributes.data.contents[i].files[0];
881
882       if(i>=0){
883         var restricaocharacter = model.title.substring(0,200); //limita os caracteres a 200
884         var restricaocharacter1 =model.lead.substring(0,60)
885         self.$("#intermedia").append('<div data-type='+model.id+' class="cols four"><article
886         }cont++;
887
888       }
889     }
890
891   },
```

Figura 36- View intermediatagsview

Na fig37 o id é enviado para a rota home/id representado na Figura 31, com o intuito de quando uma notícia é selecionada esta vai ser impressa como notícia principal.

```
911 Homepostview = Backbone.View.extend({
912     el: "#content",
913     template: "post.html",
914     template: postTpl,
915     initialize: function(options) {
916         //this.id= options.id||"";
917         console.log("idsjhdjkcfdhjkcfb",options);
918         idtes: options.id || "";
919     },
920
921     render: function() {
922         $(this.el).html(_.template(this.template));
923         console.log("idtes.....",this.idtes);
924         app.navigate('home/'+this.idtes, { trigger: true });
925     },
926
927     render2: function(x) {
928         $(this.el).html(_.template(this.template));
929         console.log("id.....",x);
930         app.navigate('home/'+x, { trigger: true });
931     },
932
933     close: function(){
934         this.remove();
935     }
936 });
937
938
```

Figura 37- View homepostview

Por fim é atribuída outra rota fig38 e repete-se todo o processo na construção de páginas.

```
138 ▼ var ApplicationRouter = Backbone.Router.extend({
139     //sao criadas as rotas
140     routes: {
141         "home": "home",
142         "home/:id": "home2",
143         "home1/:id": "home3"
144     },
145
```

Figura 38- Rotas possíveis

4.4 Controlo de versões

Uma ferramenta muito importante no decorrer do projeto foi o SourceTree representado na fig39, que pertence ao Bitbucket, com ela resolveu-se problemas relacionados com segurança, disponibilidade e mobilidade.

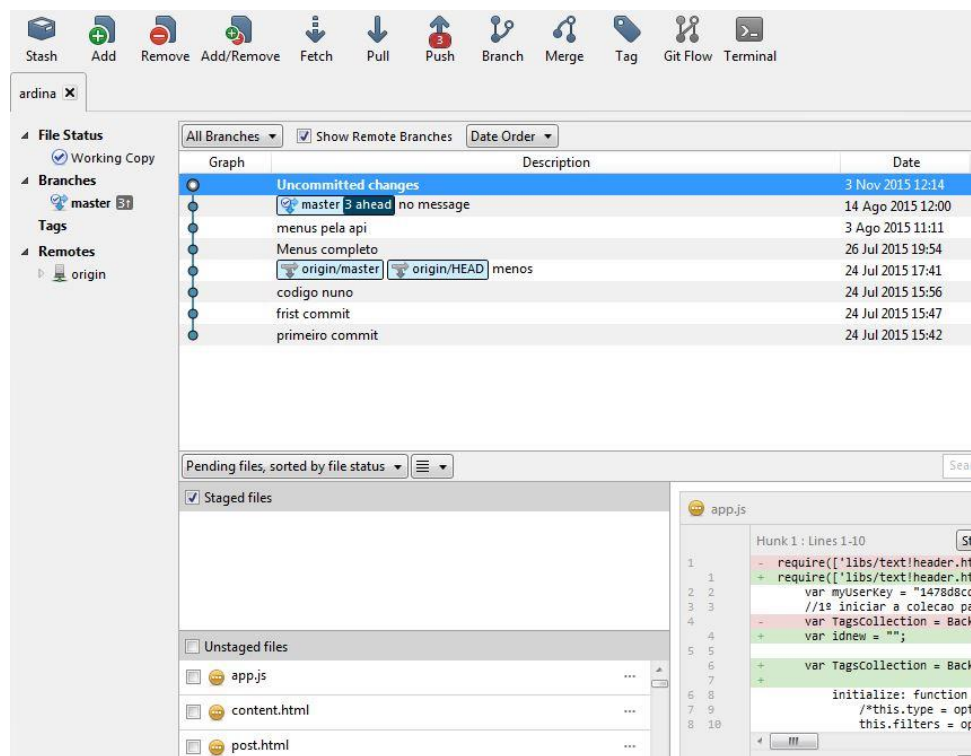


Figura 39- SourceTree controlo de versões

Segurança pois no caso de sucessivos erros na aplicação ou avaria no equipamento tem-se sempre uma versão anterior funcional. Mobilidade porque em qualquer lugar posso aceder a aplicação. Disponibilidade pela partilha do projeto com toda a equipa de trabalho.

5 Testes e Conclusão

5.1 Resultados Obtidos

Os resultados obtidos estão de acordo com o planeado inicialmente com a exceção de um ponto ou outro que inicialmente se pretendia implementar mas com o decorrer do tempo verificou-se que não era possível cumprir dentro dos prazos estipulados.

Foi contruída uma view dedicada a publicidade externa, mas com o decorrer do tempo verificou-se que já não se ia por em prática.

5.2 Testes e validações

Para testar o website usou-se o **Page Speed Insights** que é uma ferramenta online que dá a indicação numa escala de 0% a 100% da funcionalidade da aplicação e quais as melhorias a refazer (17).

Na figura 40 é possível visualizar a avaliação do website com os ditos resultados.

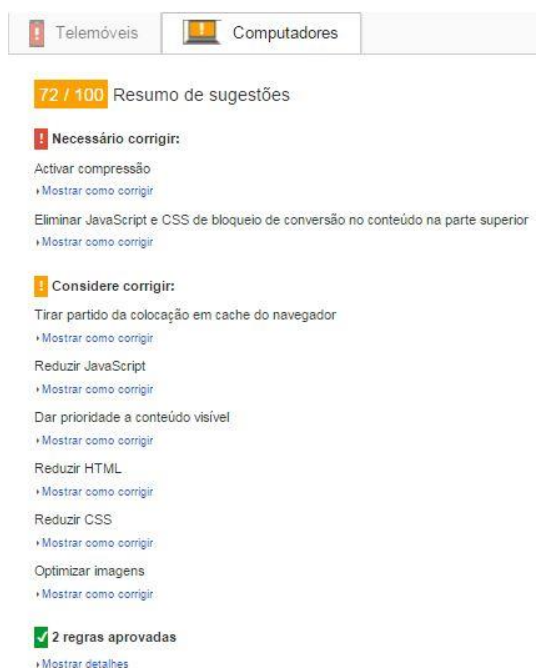


Figura 40- Teste do website

5.3 Conclusão

Inicialmente, na análise do que se pretendia por em prática surgiram algumas dúvidas, mas com o passar do tempo, com a ajuda dos colaboradores da Dom Digital e com o estudo prévio essas dificuldades foram ultrapassadas.

No início da implementação do projeto tinha em vista utilizar mais algumas funcionalidades mas optei por não por em pratica devido aos prazos de tempo que tinha.

Na fase de implementação o projeto foi sofrendo algumas alterações. Alterações essas necessárias, devido aos erros que foram aparecendo no código com o desenrolar do projeto mas também porque em algumas situações chegou-se a conclusão que essas alterações trariam benefícios.

Numa análise geral os objetivos propostos e planeados foram colocados em prática.

5.4 Implementações futuras

No futuro e com um prazo de tempo mais longo gostaria de melhorar alguns aspetos da aplicação como:

- Colocar um menu de login para os potenciais utilizadores da aplicação.
- Melhorar um pouco o aspeto da aplicação a nível de cores e contrastes.
- Colocar a aplicação online.

Referências bibliográficas

1. Novas Tecnologias de Informação. *Dom Digital*. [Online] [Citação: 2 de Junho de 2015.] <http://www.domdigital.pt/sobrenos/quemsomos.asp>.
2. Serviços-parceiro salesforce Portugal. *Dom Digital*. [Online] [Citação: 3 de junho de 2015.] <http://www.domdigital.pt/servicos/salesforce.asp>.
3. Sales Cloud. *Dom Digital-social Enterprise-sales cloud*. [Online] [Citação: 5 de junho de 2015.] <http://www.domdigital.pt/servicos/salescloud.asp>.
4. Service Cloud. *Dom Digital-Serviços-salesforce-service cloud*. [Online] [Citação: 5 de Junho de 2015.] <http://www.domdigital.pt/servicos/servicecloud.asp>.
5. Salesforce . *Dom Digital-serviços-salesforce-markting cloud*. [Online] [Citação: 5 de Junho de 2015.] <http://www.domdigital.pt/servicos/marketingcloud.asp>.
6. Community cloud. *Dom Digital.salesforce-Community cloud*. [Online] [Citação: 5 de Junho de 2015.] <http://www.domdigital.pt/servicos/communitycloud.asp>.
7. analytics cloud. *Dom Digital-serviços-salesforce-wave analytics cloud*. [Online] [Citação: 2 de Junho de 2015.] <http://www.domdigital.pt/servicos/waveanalyticscloud.asp>.
8. Desenvolvimento de Aplicações. *Dom Digital-serviços-desenvolvimento de aplicações*. [Online] [Citação: 5 de Junho de 2015.] <http://www.domdigital.pt/servicos/desenvolvimentodeaplicacoes.asp>.
9. paas. *Salesforce*. [Online] [Citação: 5 de Junho de 2015.] <https://www.salesforce.com/paas/overview/&prev=search>.
10. ardina.api. *Dom Digital- Serviços Internet*. [Online] [Citação: 2 de Junho de 2015.] <http://www.domdigital.pt/produtos/ardina.com/ardinaapi.asp>.
11. ardina.press. *Dom Digital-serviços internet-produtos*. [Online] [Citação: 5 de Junho de 2015.] <http://www.domdigital.pt/produtos/ardina.com/ardinapress.asp>.

12. Backbone.js. *Série Backbone.js: Parte 01 – Introdução*. [Online] [Citação: 6 de Junho de 2015.] <http://imasters.com.br/artigo/24577/javascript/serie-backbonejs-parte-01-introducao/>.
13. Guia para desenvolvedores Java. *Serviços de cloud computing PaaS*. [Online] [Citação: 8 de Setembro de 2015.] http://www.infoq.com/br/articles/paas_comparison.
14. backbone. *Introdução ao backbone devmedia*. [Online] [Citação: 9 de Setembro de 2015.] <http://www.devmedia.com.br/introducao-ao-backbone-js/29303>.
15. controle versões OpenSource. *Os 6 melhores control de versões open source*. [Online] [Citação: 7 de setembro de 2015.] <https://edersonmelo.wordpress.com/2008/09/22/os-6-melhores-control-versoes-opensource/>.
16. Scrum . *Metodologia scrum*. [Online] [Citação: 6 de Setembro de 2015.] <https://www.google.pt/search?biw=1280&bih=675&tbm=isch&sa=1&btnG=Pesquisar&q=metodologia+scrum#btnG=Pesquisar&imgc=46MHpuf7GXgbhM%3A>.
17. pagespeed. *PageSpeed Insights*. [Online] [Citação: 4 de julho de 2015.] <https://developers.google.com/speed/pagespeed/insights/?url=http%3A%2F%2Fwww.sal.ipg.pt%2Fuser%2F1010395%2F%23home&tab=desktop>.