



Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores

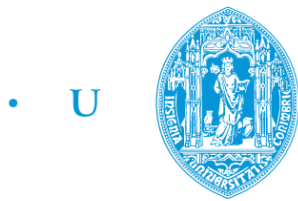
Ana Carolina Monteiro Nujo

SmartCity Coimbra Dashboard

Setembro 2015



UNIVERSIDADE DE COIMBRA



• C •

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Eletrotécnica e de Computadores
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

SmartCity Coimbra Dashboard

Autor:

Ana Carolina Monteiro Nujo

Membros do Júri:

Presidente: Humberto Manuel Matos Jorge

Orientador: António Manuel Oliveira Gomes Martins

Vogal: Rita Cristina Girão Coelho da Silva

Setembro de 2015

Agradecimentos

Agradeço ao Professor Doutor António Manuel de Oliveira Gomes Martins pela orientação prestada, disponibilidade, sugestões pertinentes e sobretudo ensinamentos transmitidos ao longo de toda a realização deste trabalho.

Ao meu orientador de estágio, o Engenheiro Francisco Maia, e a toda a equipa da Streamline, pelo seu apoio e orientação disponibilizados na realização deste trabalho, conselhos e sugestões.

Agradeço a todas as entidades e empresas que receberam com entusiasmo a ideia deste projeto e por toda a disponibilidade mostrada: EDP Distribuição, Águas do Mondego, Lusitaniagás, Sistema Nacional de Informação de Recursos Hídricos, Divisão do Ambiente do Departamento de Desenvolvimento Social e Ambiente, Comissão de Coordenação e Desenvolvimento Regional do Centro, Agenda7 Coimbra e ao Engenheiro João Carrilho do DEM.

Agradeço à minha família principalmente aos meus pais, irmão e cunhada, pelo grande carinho e apoio incondicional durante todo este tempo. Um agradecimento especial ao meu afilhado Afonso cujo nascimento a meio do desenvolvimento deste projeto me deu motivação extra.

Às minhas amigas, Carol, Cris, Diana, Maria João, Marina, Neia e Pita, agradeço a compreensão e o carinho mostrado neste percurso, sem elas não teria sido a mesma coisa.

Finalmente, agradeço a todos os meus colegas de curso, em especial à Lola, Caseiro, Jorge, Tiago, Sampaio e Sérgio, por sempre me fazerem acreditar em mim e por terem tornado o meu percurso académico mais rico.

A todos o meu obrigado.

Ana Nujo

Resumo

Numa tentativa de integrar Coimbra no movimento smart cities, o projeto SmartCity Coimbra foi lançado em 2012 com o intuito de mobilizar os cidadãos e os agentes económicos para a adoção de soluções inovadoras para as questões que se colocam à gestão urbana em áreas como a energia, mobilidade, saúde, turismo e governo local. Uma das ferramentas mais visíveis e dinâmicas de um projeto Smart City é o seu dashboard, uma plataforma online com vários indicadores e informações que se destinam a tornar acessível aos cidadãos (residentes e visitantes) dados relevantes sobre o uso que a cidade faz dos recursos, numa perspetiva simultaneamente de utilidade e de sensibilização para a sustentabilidade.

O primeiro protótipo do SmartCity Coimbra foi desenvolvido no passado ano letivo, ainda que com muitas limitações, exibindo informações relativas a meteorologia, transportes públicos, qualidade do ar, promoção turística, notícias e agenda de eventos culturais.

Esta dissertação tem como objetivo reconstruir o primeiro protótipo do SmartCity Coimbra Dashboard, acrescentando informação considerada relevante, nomeadamente o consumo de diferentes recursos: Energia Elétrica, Água e Gás. Pretende também reformular o seu *design* para um mais apelativo, concretizando ligação de dados com várias fontes e efetuando o respetivo tratamento e afixação no Dashboard, implementando um conjunto de funcionalidades que não existiam e/ou não estavam previstas, tendo em vista a integração no futuro website SmartCoimbra.

Palavras-chave: Smart City, Web Dashboard, SmartCoimbra, CakePHP, MySQL

Abstract

In an attempt to instate Coimbra in the smart cities movement, the SmartCity Coimbra project was released in 2012 with the ambition of mobilizing the citizens and economic agents to adopt innovative solutions to deal with the urban management of energy supply, mobility, health care, tourism and local government. One of the most widely known tools of a Smart City project is the dashboard, an online platform that gathers multiple indicators and information in a single site, meant to engage the citizens (locals and visitors) in the management of the city resources, aspiring to give a helpful insight as well as sensitize to the sustainability issue.

The first prototype of the SmartCity Coimbra was developed last year, though with significant limitations, providing a small set of information regarding, e.g. weather forecast, power consumption, public transportation, air quality, as well as tourism promotion, news and cultural agenda.

The present master's dissertation aims at rebuilding the first SmartCity Coimbra Dashboard prototype, adding relevant information such as the consumption of different resources: Electric energy, Water and Gas. It also intends to evolving its design, accomplishing the connection to the different data sources, processing its data and presenting it in the Dashboard, implementing a set of features that were not present and/or were not foreseen, with the intent of being integrated in the future SmartCoimbra website.

Keywords: Smart City, Web Dashboard, SmartCoimbra, CakePHP, MySQL

Índice

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Índice de Figuras	ix
Índice de Tabelas	x
Abreviaturas	xi
Guia-Síntese de Definições.....	xiii
1. Introdução	1
1.1 SmartCity Coimbra.....	2
1.2 SmartCity Coimbra Dashboard	2
1.3 Ponto de Partida.....	3
1.4 Motivação e objetivos para a evolução.....	9
2. Especificação e organização do Dashboard	10
2.1 Blocos existentes	10
2.2 Blocos a acrescentar	14
3. Identificação dos produtores de informação e especificação dos conteúdos	17
3.1 Meteorologia.....	17
3.2 Nível do Rio.....	18
3.3 Qualidade do Ar.....	19
3.4 Energia Elétrica	19
3.5 Água.....	20
3.6 Gás	21
3.7 Consumo de Recursos	21
3.8 Produção de Resíduos.....	22
3.9 Mobilidade.....	23
4. Escolha das ferramentas de suporte e construção do Dashboard	24
4.1 Bootstrap.....	26
4.2 Base de Dados	27
4.3 CakePHP.....	28
4.4 AngularJs.....	30

4.5	Cron Job.....	34
4.6	Gestão do desenvolvimento.....	35
5.	Conteúdo do Dashboard.....	37
5.1	Painel Meteorologia.....	37
5.2	Painel Nível do Rio.....	39
5.3	Painel Qualidade do Ar.....	40
5.4	Painéis Energia Elétrica, Água e Gás	42
5.5	Painel Consumo de Recursos	47
5.6	Painel Produção de Resíduos.....	48
5.7	Painel Emissões de CO ₂	49
5.8	Painel Mobilidade.....	50
5.9	Painel Eventos	51
5.10	Painel Locais de Interesse.....	51
5.11	Painel Notícias.....	52
5.12	Painel Farmácias.....	53
5.13	Modal.....	54
5.14	Versão Final.....	54
5.15	Versão Mobile	56
6.	Conclusão e sugestões de trabalhos futuros.....	58
	Referências.....	61
	Apêndice A – Manual técnico de Manutenção	63
	Apêndice B – Cálculo do Índice de Qualidade do Ar.....	77
	Apêndice C – Textos Informativos	83
	Anexo A – Protocolo e Anexo Técnico.....	97

Índice de Figuras

Fig. 1 - Aspeto final do protótipo SmartCity Coimbra Dashboard	8
Fig. 2 - Página “Acerca” do Protótipo do Smart City Coimbra Dashboard.....	9
Fig. 3 - Parte da informação disponível no website Weather Underground	11
Fig. 4 - Mouseover sobre a agenda do “Agenda 7 Coimbra”	12
Fig. 5 - Pesquisador de itinerários da TRANSDEV	13
Fig. 6 - Pesquisador de itinerários da CP	13
Fig. 7 - Caudal afluente médio diário da estação Açude Ponte.....	15
Fig. 8 - Parte da informação disponibilizada no website Farmácias de Serviço.....	16
Fig. 9 - Fase inicial de construção.....	27
Fig. 10 - Esquema da gestão e desenvolvimento do Dashboard	35
Fig. 11 - Representação dos comandos usados em Git Bash	36
Fig. 12 - Painel Meteorologia.....	39
Fig. 13 - Painel Nível do Rio.....	39
Fig. 14 - Painel Qualidade do Ar.....	42
Fig. 15 - Painéis Energia Elétrica, Água e Gás	44
Fig. 16 - Painel Consumo de Recursos	48
Fig. 17 - Painel Produção de Resíduos.....	49
Fig. 18 - Painel Emissões de CO ₂	50
Fig. 19 - Painel Mobilidade.....	51
Fig. 20 - Painel Eventos	51
Fig. 21 - Painel Locais de Interesse.....	52
Fig. 22 - Painel Notícias.....	53
Fig. 23 – Painel Farmácias	54
Fig. 24 – Aspeto final do Dashboard.....	55
Fig. 25 – Ligações entre o Dashboard e as várias fontes de informação	56
Fig. 26 - Versão mobile.....	57
Fig. 27 - Estrutura de arquivos do CakePHP	61
Fig. 28 - Funcionamento da arquitetura MVC	63
Fig. 29 - Estrutura de arquivos do AngularJs.....	69

Índice de Tabelas

Tabela 1 - Tipos de parâmetros de informação disponível em cada dashboard.....	4
Tabela 2 - Classificação do Índice de Qualidade do Ar proposto para o ano 2015, APA. Valores em $\mu g/m^3$	78
Tabela 3 - Escala de índice UV estabelecido pela EPA (United States Environmental Protection Agency).....	81
Tabela 4 - Classificação do Índice de Qualidade do Ar proposto para o ano 2015, APA. Valores em $\mu g/m^3$	83
Tabela 6 - Conselhos de saúde em função do índice IQar, APA.....	84
Tabela 7 - Infraestruturas do Município de Coimbra.....	87
Tabela 8 - Emissões de CO ₂ das viaturas de recolha de resíduos.....	88

Abreviaturas

APA	Agência Portuguesa do Ambiente
BD	Base de Dados
CCDRC	Comissão de Coordenação e Desenvolvimento Regional do Centro
CO	Monóxido de Carbono
CO ₂	Dióxido de Carbono
CP	Comboios de Portugal
CSS	<i>Cascading Style Sheets</i>
DEEC	Departamento de Engenharia Eletrotécnica e de Computadores
DEM	Departamento de Engenharia Mecânica
DOM	<i>Document Object Model</i>
EDP	Energias de Portugal
ERSE	Entidade Reguladora dos Serviços Energética
FTP	<i>File Transfer Protocol</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEA	<i>International Energy Agency</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model, View e Controller</i>
NO ₂	Dióxido de azoto
PHP	<i>Personal Home Page</i>
PM10	Partículas inaláveis de diâmetro inferior a 10 micrómetros (µm)
PRM	Posto de Regulação e Medição

REN	Rede Elétrica Nacional
RSS	<i>Really Simple Syndication</i>
SMTUC	Serviços Municipalizados de Transportes Urbanos de Coimbra
SNIRH	Sistema Nacional de Informação de Recursos Hídricos
SO ₂	Dióxido de enxofre
SSH	<i>Secure Shell</i>
REEE	Resíduos de Equipamentos Elétricos e Eletrónicos
SQL	<i>Structure Query Language</i>
TIC	Tecnologias da Informação e Comunicação
UC	Universidade de Coimbra
URL	<i>Uniform Resource Locator</i>
XML	<i>Extensible Markup Language</i>

Guia-Síntese de Definições

AngularJs	Framework JavaScript de código aberto, mantido pela Google, que auxilia na execução de aplicações.
Array	Variável que armazena vários elementos do mesmo tipo que podem ser identificados por um índice.
CakePHP	Framework completo de código aberto escrito em PHP que oferece uma estrutura que possibilita aos programadores de PHP de todos os níveis desenvolver aplicações robustas rapidamente.
cURL	Utilitário e biblioteca que auxilia na procura de informações de outros <i>websites</i> , possibilitando a obtenção de dados remotos.
Cron Job	Sistema Unix que permite o agendamento de operações que funcionam em horários pré-estabelecidos e datas agendadas.
Feed	É um formato de dados que representa o conteúdo de um <i>website</i> ou aplicação que pode ser exibida em outros <i>websites</i> e ferramentas.
Git	Sistema de Controlo de versões e de gestão de código fonte.
Git BASH	Ferramenta Windows usada para executar o comando Git na linha de comandos.
GitLab	Interface web usada para comunicar com o Git.
Highcharts	Biblioteca de gráficos escrita em JavaScript.
iFrame	<i>Inline Frame</i> . Mecanismo que permite a inserção de um documento HTML embutido noutro documento HTML ou website.
Postman	Aplicação do Google Chrome usada para interagir com APIs (Application Program Interface) em HTTP. Permite construir pedidos e ler respostas
JavaScript	Linguagem de programação do lado do cliente orientada a objetos.
JQuery	Biblioteca em JavaScript contendo um pacote de funções que simplificam e estendem a linguagem mãe.
MySQL Workbench	Ferramenta visual para <i>design</i> , desenvolvimento e administração de base de dados MySQL.
NetBeans	<i>Software</i> de código aberto, disponível para Windows, Linux e Mac, que permite desenvolver páginas <i>web</i> e aplicações para dispositivos móveis.
PuTTY	<i>Software</i> de emulação de terminal grátis e de código livre que permite realizar ligações com servidores remotos pela linha de comandos.
Template	Documento de conteúdo que contém apenas a apresentação visual e instruções sobre onde e qual tipo de conteúdo deve entrar a cada parcela da apresentação.
Twitter Bootstrap	Framework criado pela equipa do Twitter, de código aberto, para facilitar o desenvolvimento de <i>websites</i> e sistemas <i>web</i> .
Tooltip	Conhecido como Dica de Contexto, é um texto informativo que aparece ao passar o ponteiro do rato sobre um elemento.
Widget	Componente usado para simplificar o acesso a outro programa ou sistema. A sua interface gráfica permite adicionar tarefas específicas à aplicação.
XAMPP	Pacote combinado com os principais servidores de código aberto do mercado, incluindo FTP, base de dados MySQL e Apache com suporte as linguagens PHP e Perl.

1. Introdução

As novas Tecnologias de Informação e Comunicação (TIC) vieram alterar por completo a forma como as interações se dão em sociedade, seja no ambiente doméstico, trabalho, saúde, segurança ou transporte. O seu impacto levou à procura de soluções inovadoras para a gestão inteligente e desenvolvimento sustentável dos nossos meios urbanos. Neste contexto surgiram as Smart Cities, cidades inteligentes onde as novas tecnologias se traduzem em melhores serviços públicos para os cidadãos, melhor uso dos recursos disponíveis e menor impacto no ambiente.

Num projeto conduzido em 2007 pelo Centro de Ciência Regional da Universidade de Tecnologia de Viena (Áustria), Coimbra foi a única cidade portuguesa ponderada com potencial para ser uma Smart City. Neste estudo uma cidade é considerada inteligente se apresenta um bom desempenho segundo seis dimensões urbanas (Economia, Mobilidade, Ambiente, Cidadãos, Vida e Governação), que se entendem como fundamentais para um efetivo desempenho inteligente. Para além de Coimbra, outras 70 cidades foram selecionadas de um universo inicial de 1600 cidades europeias candidatas. Os três requisitos obrigatórios para a seleção foram:

- População urbana entre 100 000 e 500 000 habitantes;
- Pelo menos uma universidade;
- Área de influência regional com menos de 1,5 milhão de habitantes.

O objetivo do projeto é modelar a cidade na forma como usa as novas tecnologias em prol dos seus cidadãos e das empresas locais, nomeadamente nas áreas da energia, mobilidade, saúde, turismo e governo local. Tudo isto tendo em conta o espírito próprio e muito especial que Coimbra possui, caracterizado pelo seu património cultural e ecológico, gentes e tradições.

Esta dissertação trata do desenvolvimento de uma plataforma *online* onde se encontrem reunidas as informações e indicadores de maior importância da cidade de Coimbra, a “SmartCoimbra Dashboard”. Esta plataforma visa manter informados cidadãos e turistas das principais características da cidade, os seus consumos e impacto ambiental para ajudar a melhorar a qualidade de vida dos mesmos e envolvê-los ativamente no rumo e evolução da mesma.

1.1 SmartCity Coimbra

O conceito de SmartCity continua a ser algo vago e impreciso, mas as cidades inteligentes são vistas por muitos como uma abordagem ampla e integrada para melhorar a eficiência da gestão urbana, a qualidade de vida dos cidadãos e a crescente economia local. Em suma, uma SmartCity procura formas de otimizar as suas infraestruturas sem perder de vista as necessidades dos seus cidadãos [1][2][3].

O termo está em voga desde os anos 90 mas foi nos últimos anos, em parte devido à severa crise económica, em parte para mitigar os problemas gerados pelo crescimento da população urbana, que este conceito foi desenvolvido aprofundadamente. As alterações climáticas e escassez de recursos também contribuíram para o seu desenvolvimento. No final da década passada, o conceito já tinha cativado a imaginação de várias nações em todo o mundo.

O projeto SmartCity de Coimbra pretende contribuir para tornar Coimbra uma cidade modelar na forma como usa as novas tecnologias em prol dos seus cidadãos e das empresas locais, nomeadamente nas áreas da energia, mobilidade, saúde, turismo e governo local. Para tal, pretende atuar em cinco áreas Smart: Saúde, Energia, Mobilidade Inteligente, Turismo e Autarquia Inteligente. Este projeto depende da colaboração e parceria com diversas entidades, entre as quais a Universidade de Coimbra, responsável pelo lançamento do SmartCity Coimbra Dashboard.

1.2 SmartCity Coimbra Dashboard

Uma das ferramentas mais visíveis e dinâmicas de um projeto Smart City é o seu Dashboard. Um Dashboard pode ser definido como um painel informativo onde se encontram reunidos indicadores e informações caracterizadores da cidade e do uso que esta faz de um conjunto relevante de recursos. Estas informações são maioritariamente relacionadas com os serviços públicos mais relevantes e são apresentadas com a máxima frequência de disponibilização usada pelos respetivos produtores, de forma a permitir uma monitorização constante das mesmas. Esta característica dinâmica dos Dashboard permite também envolver os cidadãos de forma ativa na evolução da sua cidade ao ajudar na tomada de decisões conscientes e informadas [4].

O Dashboard Coimbra terá de se adaptar à cidade e às necessidades dos seus cidadãos, pelo que o conjunto dos indicadores e possíveis produtores de dados será estudado e escolhido cuidadosamente, tendo como base quatro das cinco áreas do projeto SmartCoimbra: Saúde, Energia, Mobilidade Inteligente e Turismo.

1.3 Ponto de Partida

O presente projeto, Smart Coimbra Dashboard é uma continuação do projeto iniciado no passado ano letivo pelo colega Hugo Ramalho. O produto intermédio que funciona como ponto de partida da presente dissertação é um protótipo de painel de monitorização da cidade de Coimbra constituído por um conjunto de informações respeitantes ao funcionamento da cidade de Coimbra, sendo estas relativas ao turismo, à mobilidade, à meteorologia, à energia elétrica, à cultura, às notícias e ao ambiente. Para compreender melhor o que levou a escolher estas informações em detrimento de outras e o estado parcialmente ativo de algumas delas, segue-se uma síntese do processo de desenvolvimento deste primeiro protótipo do Dashboard.

1.3.1 Levantamento de Projetos Smart City Dashboard

O primeiro passo foi o levantamento e caracterização de outros projetos Smart City Dashboard existentes *online*, de forma a retirar algumas conclusões e ideias passíveis de serem utilizadas na conceção do “Smart Coimbra Dashboard”. Foi feita uma pesquisa de *web dashboard*, tendo sido analisadas para este projeto as seguintes:

- Oberlin Environmental Dashboard[5]
- London Dashboard[6]
- City Dashboard Amsterdam[7]
- Birmingham, Brighton, Cardiff, Edinburgh, Glasgow, Leeds e Manchester Dashboard
- Green Addict Bristol[8]
- Follow Aarhus[9]
- Green Energy Dashboard[10]
- Facebook Dashboard

1.3.2 Sistematização da informação recolhida no levantamento

Feito o levantamento dos *web Dashboards* existentes, prosseguiu-se com a recolha dos dados empregados em cada painel de informação. O cruzamento destes dados permitiu aferir que os parâmetros mais utilizados em cada *web dashboard* são a Meteorologia, Energia elétrica consumida, Qualidade do ar (CO₂), Notícias e Mapa. Por não existir informação disponível sobre a construção dos dashboards não foi possível descobrir a fonte dos vários parâmetros disponíveis ou a forma como foram recolhidos. Ainda assim, foi possível observar quais os dados habitualmente com mais significado na construção de um dashboard (Tabela 1) e que se devem encontrar presentes no protótipo de Dashboard a construir para a cidade de Coimbra.

Parâmetros Dashboard	Meteorologia				Transportes Públicos			Eletricidade	Qualidade do Ar	Cameras de Trânsito	Notícias	Água	Mapa	Outros	
	Estado do Tempo	Temperatura	Vento	Humidade	Precipitação	Bicicletas	Metro	Autocarros		Concentrações		Consumida		Satisfação dos cidadãos	Natalidade /Mortalidade
Oberlin Environmental Dashboard		x	x	x					x	CO ₂		x			
London Dashboard	x	x	x	x	x	x	x	x	x	Vários	x	x	x	x	
Dashboard Amsterdam							x	x		Vários			x		
Birmingham Dashboard	x	x	x							Vários	x		x	x	
Green Addict Bristol									x	CO ₂			x		
Follow Aarhus	x	x	x	x	x		x	x		CO, NO ₃		x		x	x
Green Energy dashboard		x							x	CO ₂					
Facebook Dashboard		x		x					x			x			

Tabela 1- Tipos de parâmetros de informação disponível em cada dashboard

1.3.3 Análise dos parâmetros para o Dashboard de Coimbra

Nesta fase foram escolhidas em definitivo quais as informações a serem disponibilizadas no Dashboard. Todos os parâmetros mais comuns nos *web* dashboard existentes foram considerados importantes para integrar o Dashboard Coimbra, embora sujeitos a alterações para se ajustarem à realidade e à dinâmica de Coimbra.

A informação relativa aos transportes públicos foi considerada importante para os cidadãos, tendo em vista a programação de itinerários por partes destes, e foi portanto incluída. Após uma consulta de informações *online* disponíveis relativa à cidade de Coimbra, optou-se por incluir também uma agenda de eventos, já previamente criada, contendo os eventos culturais e científicos a realizar em cada dia em Coimbra.

De forma a quantificar o impacto ambiental foi decidido adicionar um novo parâmetro, as Emissões de CO₂. Estas emissões são estimadas com base na informação disponível sobre o mix de produção que satisfaz a procura nacional de energia elétrica, multiplicando os valores da produção de energia elétrica pelos fatores de emissão, para cada uma das fontes de energia (gás natural ou carvão).

Após a seleção dos parâmetros desejados para inclusão no Dashboard, foram averiguadas as possíveis fontes de informação dos mesmos na cidade de Coimbra. A seleção final teve em conta fatores de decisão como a facilidade de obtenção dos dados, a relação de proximidade de produtores de informação, a periodicidade com que os dados são obtidos e os requisitos tecnológicos. Alguns parâmetros encontram-se disponíveis *online* ao passo que outros têm de ser disponibilizados pelo produtor. O passo seguinte foi o contato com os interlocutores, quando

necessário, e averiguar a sua disponibilidade para participar no projeto e fornecer os valores para publicação no Dashboard.

1.3.4 Desenvolvimento do protótipo

Dado que o objetivo final era a integração do protótipo de Dashboard na página *web* do projeto principal “SmartCoimbra”, a escolha das ferramentas para o seu desenvolvimento foi feita de forma a facilitar a integração no projeto mencionado.

Foi usado o “CakePHP”, um *framework* de desenvolvimento rápido para PHP, livre e de código aberto, na versão 2.4.7. A base de dados selecionada para armazenar as informações essenciais ao Dashboard foi a MySQL versão 6.0. Esta base de dados foi escolhida em conformidade com o *framework* CakePHP, pois esta é a mais utilizada nas aplicações *web* baseadas na linguagem PHP. O *software* usado na administração da BD é o MySQL Workbench. Nesta BD foram criadas três tabelas para armazenar os dados alusivos à meteorologia, ao consumo de energia e à qualidade do ar.

Na conceção da primeira versão do protótipo foram usadas várias linguagens, como SQL, PHP, HTML e CSS. SQL é uma linguagem usada para implementar a gestão do conteúdo armazenado na base de dados, PHP encarrega-se de toda a lógica da aplicação *web*, o HTML é usado para estruturar os conteúdos da página, enquanto o CSS é utilizado para formatar conteúdos estruturados da página.

1.3.5 Conteúdos do Dashboard

Uma vez reunidos todos os parâmetros e valores previamente selecionados para entrarem no Dashboard, os conteúdos finais em cada bloco a serem apresentados foram os seguintes:

Bloco Meteorologia:

O bloco ficou composto por: Temperatura ambiente, Estado do tempo, Humidade, Velocidade do vento, Previsão de Temperatura Máxima, Previsão de Temperatura Mínima e Imagem de Previsão de Estado de Tempo. A informação era obtida através do *feed* Yahoo! Weather e depois guardada numa tabela relativa à meteorologia, na base de dados deste projeto.

A escolha deste *feed* deve-se à disponibilização de todos os dados requeridos para apresentar no Dashboard, para além de ser completamente gratuita.

Bloco Notícias:

Este conteúdo era fornecido pelos *feeds* do jornal online “Notícias de Coimbra” e do *website* “Destakes” pois estes providenciam notícias com uma boa atualização diária sobre a região de Coimbra. A informação figura no primeiro protótipo da seguinte forma: Título da notícia, Texto parcial da notícia e *link* para a fonte (local *web*) da respetiva notícia. O conteúdo era apresentado

dentro de um bloco, onde era possível fazer *scroll* e navegar entre as várias notícias. Apesar de estes elementos poderem ser obtidos e guardados numa tabela na base de dados do projeto, este não foi o método escolhido. Foi usada a ferramenta gratuita *online* “Feed Wind”, que converte um ou mais *feed*'s RSS num *widget*, que pode ser facilmente colocado num *website* por meio de código embutido. A frequência de atualização desta aplicação era de quinze minutos.

Bloco Energia Elétrica Consumida:

A energia elétrica consumida em Coimbra é fornecida por três subestações: Alegria, Relvinha e Alto de S. João. As potências instaladas nas três subestações são respetivamente 60 MVA, 80 MVA e 40 MVA. Os valores dos consumos seriam facultados pela “EDP Distribuição” mas à data da construção do primeiro protótipo Dashboard a ligação à base de dados da EDP não foi conseguida. Por essa razão foram introduzidos manualmente valores na BD correspondentes a um dia, com periodicidade de 15 minutos, num total de 96 entradas.

Este bloco mostrava a energia elétrica consumida atual e ainda um diagrama de carga referente a um dia completo, ou seja vinte e quatro horas. Este tipo de diagrama traduz a variação da potência do consumo ao longo das horas e foi conseguido com a ferramenta “GoogleCharts”.

Bloco Agenda:

A agenda tem uma organização mensal, na qual estão marcados os eventos (atividade cultural da cidade de Coimbra) a decorrer em cada dia do mês. A agenda tem como origem o *website* “Agenda7-Coimbra” (agenda7.uc.pt/agenda7), fruto de um projeto da Universidade de Coimbra com uma parceria com a Câmara Municipal de Coimbra. No seu *website* é facultada o *iframe* da agenda que foi embutida no primeiro protótipo.

Bloco Qualidade do Ar:

Neste bloco era apresentada uma tabela com os valores de concentração dos variados poluentes atmosféricos (Ozono, Dióxido de Azoto, Monóxido de Carbono, Dióxido de Enxofre e material particulado com dimensão inferior a 10um), recolhidos em cada uma das duas estações de monitorização da qualidade do ar instaladas na cidade de Coimbra (Avenida Fernão de Magalhães e Instituto Geofísico). Estes valores são facultados pela Comissão de Coordenação e Desenvolvimento Regional do Centro, CCDRC.

Até ao momento da construção do primeiro protótipo não tinha sido estabelecida ligação à base de dados do CCDRC ou da Agência Portuguesa do Ambiente pelo que em alternativa os valores eram lidos de um *iframe* disponibilizado pelo *website* da APA.

Bloco Locais de Interesse:

Para a construção deste bloco foi feita uma recolha de locais com interesse turístico com base nos *websites* “Turismo de Coimbra” (turismodecoimbra.pt) e “Turismo do Centro” (turismodocentro.pt). Estes pontos foram assinalados num mapa criado com a ferramenta “Google Maps Engine”. Esta ferramenta ajuda a conceber mapas personalizados e avançados para partilhar e publicar na *web*, para uso não comercial.

Bloco Emissões de CO₂:

O teor deste bloco era constituído pelas emissões estimadas de dióxido de carbono referentes à produção de energia elétrica que abastece o consumo da cidade, mais precisamente na produção não renovável respeitante à queima de carvão e de gás natural. Para estimar estas emissões foi necessário conhecer os valores de repartição de produção de energia elétrica em Portugal e ainda os fatores de emissão de CO₂ (g/kWh). Segundo os Dados Técnicos, publicados pela REN com divulgação dos principais indicadores estatísticos dos setores da sua atividade em 2013, 22% e 14% da produção total de energia elétrica correspondem à queima de carvão e de gás natural. Os fatores de emissão utilizados foram os divulgados no documento “Recomendação sobre Rotulagem de Energia Elétrica” de Dezembro de 2011, disponibilizado pela ERSE. Segundo este documento, o fator para o carvão é 973 g/kWh e para o gás natural 350 g/kWh.

As emissões de CO₂ são determinadas pela multiplicação dos valores da produção de energia elétrica pelos fatores de emissão, para cada uma das fontes de energia, conforme a expressão seguinte:

$$\text{Emissões de CO}_2 = \text{Energia elétrica produzida} * \text{Fator de emissão de CO}_2 \text{ [g]}$$

Com esta expressão, e usando os valores de potência de consumo (adquiridos num intervalo de quinze minutos) que se encontram guardados na base de dados, têm-se as expressões abaixo.

$$\text{Emissões de CO}_2 | \text{Carvão} = (\text{Potência de consumo} * 0,25 \text{ horas}) * 0,22 * 973 \text{ [g]}$$

$$\text{Emissões de CO}_2 | \text{Gás Natural} = (\text{Potência de consumo} * 0,25 \text{ horas}) * 0,14 * 350 \text{ [g]}$$

No bloco desenvolvido surgiram quatro valores: o valor das emissões correntes totais (emissões de CO₂ de queima de carvão e gás natural), dois valores das emissões repartidas por cada fonte de energia (carvão e gás natural) e emissões acumuladas nas últimas 24 horas. Foi também adicionado um *link* que redirecionava o utilizador para uma página com mais informações relativas à obtenção destas estimativas. Nesta página era explicado este método de cálculo e era, também, feita a referência aos documentos de domínio público onde podem ser consultados os valores anteriormente mencionados.

Bloco Programador de Percursos:

O conteúdo deste bloco funcionava como uma pesquisa de itinerário (autocarros a usar), redirecionando o utilizador para uma pesquisa mais avançada no *website* dos SMTUC, onde é possível visualizar o trajeto (smtuc.pt/geral/index_itinerarios.php). A integração desta pesquisa foi feita, mais uma vez, através do uso de um *iframe*. Existiam também duas hiperligações (Horários e Smtuc Mobile), que redirecionavam o utilizador para as páginas dos SMTUC relativamente aos horários de cada linha e à aplicação para *smartphone* dos SMTUC.

1.3.6 Aspeto Final do Primeiro Protótipo

A figura 1 mostra a imagem da aparência geral do primeiro protótipo dashboard.

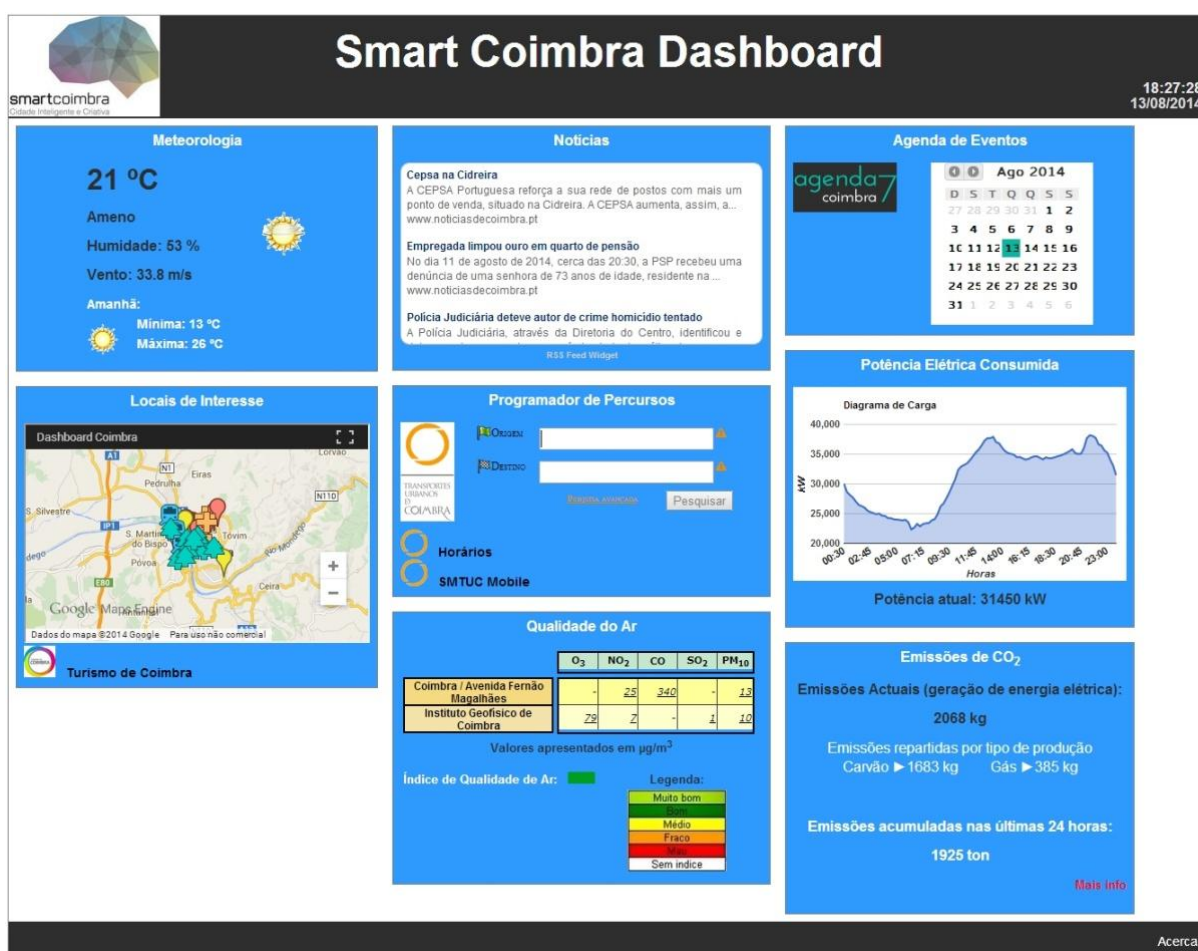


Figura 1 – Aspeto do primeiro protótipo SmartCity Coimbra Dashboard

No canto inferior direito foi colocado um *link* que redirecionava para uma página designada “Acerca”. Na página foram colocadas informações pertinentes sobre os responsáveis e as finalidades deste projeto, assim como a lista dos fornecedores de dados. É também deixado um contato, o *email* criado especificamente para este projeto, smartcoimbra-dashboard@gmail.com.

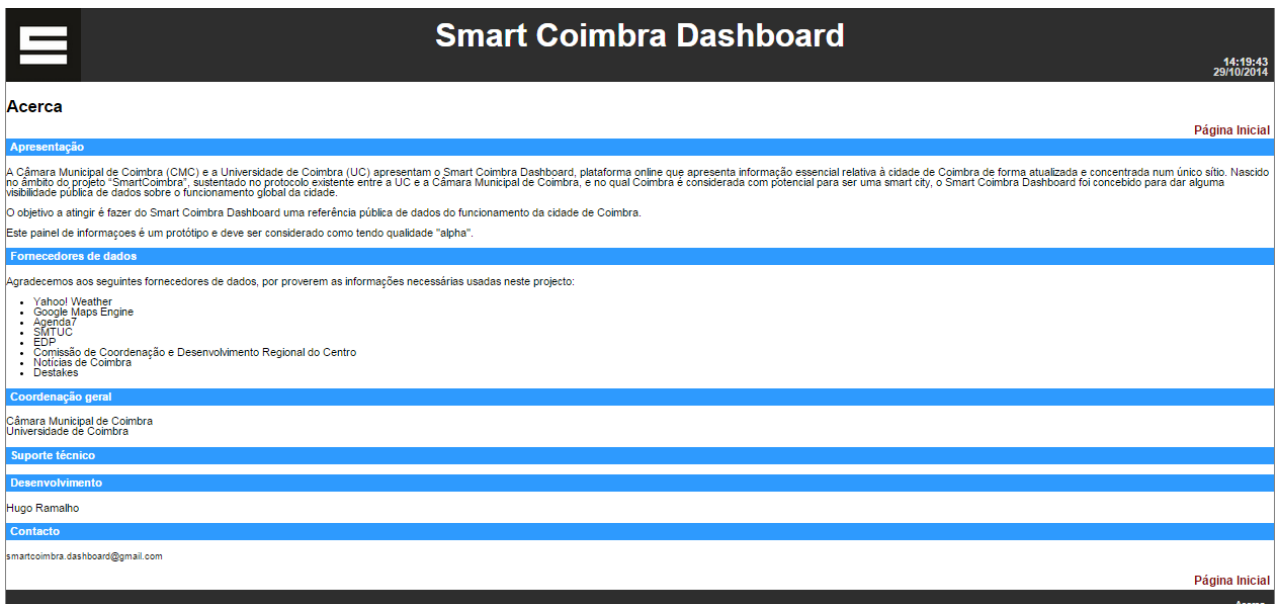


Figura 2 – Página “Acerca” do primeiro protótipo do Smart City Coimbra Dashboard

1.4 Motivação e objetivos para a evolução

Este projeto foi iniciado no ano letivo 2013/2014 com o desenvolvimento do primeiro protótipo do SmartCoimbra Dashboard. Esta dissertação tem como objetivo reconstruir o primeiro protótipo, acrescentando informação considerada relevante, reformulando o *design* para um mais apelativo, concretizando ligação de dados com várias fontes e efetuando o respetivo tratamento e afixação no Dashboard, implementando um conjunto de funcionalidades que não existiam e/ou não estavam previstas, tendo em vista a integração no futuro site SmartCoimbra.

2. Especificação e organização do Dashboard

Após uma análise detalhada do desenvolvimento do primeiro protótipo SmartCity Coimbra Dashboard, iniciou-se o processo de aperfeiçoamento e conclusão do mesmo, com vista ao desejado lançamento *online* no final desta dissertação.

A primeira etapa consistiu na pesquisa e averiguação de novos conjuntos de parâmetros suscetíveis de serem acrescentados ao Dashboard final, assim como formas de melhorar os blocos de informação já existentes. Começando por uma análise crítica ao trabalho deixado, todos os conteúdos dos blocos foram analisados para garantir que a informação e/ou fonte de informação apresentada por estes era a melhor a mais fiável entre os possíveis produtores de informação.

2.1 Blocos existentes

As conclusões tiradas da análise dos blocos de informação existentes são exibidas neste subcapítulo, onde se descrevem as possíveis alterações e ações a tomar de modo a completá-los e melhorá-los o mais possível, quando praticável.

Bloco Meteorologia:

O conteúdo deste bloco é obtido através do Yahoo! Weather. Num primeiro passo é lido do *feed* disponibilizado os campos requeridos, que são depois guardados na tabela *meteorologias* na base de dados. Embora este processo seja simples e com resultados satisfatórios, é do interesse do Dashboard Coimbra usar fontes mais próximas de Coimbra e que, por essa razão, forneçam informação mais precisa e detalhada.

Existe no Departamento de Mecânica da Universidade de Coimbra uma estação meteorológica. Nela são medidas todas as informações necessárias a este bloco, com uma periodicidade de recolha de informação na ordem dos segundos. Para além da proximidade, o facto de se tratar de um projeto da própria Universidade de Coimbra é um fator determinante da escolha do mesmo, uma vez que um dos propósitos do SmartCoimbra é envolver as entidades e empresas locais.

Os valores recolhidos por essa estação estão disponíveis no *website* Weather Underground, ao seleccionar Coimbra como o local e ADAI@DEM.UC como a estação meteorológica pretendida.



Figura 3 - Parte da informação disponível no *website* Weather Underground.

Bloco Notícias:

A informação mostrada neste bloco é retirada do jornal *online* “Notícias de Coimbra” e do *website* “Destakes”. Em ambos os casos, as notícias são obtidas através dos respetivos *feeds* disponibilizados. Por se tratar de interlocutores regionais e com periodicidade diária, estes são a melhor fonte para o nosso projeto e portanto não serão substituídos.

Bloco Energia Elétrica Consumida:

O bloco Energia Elétrica Consumida do protótipo não recolhe dados em tempo real. Mostra apenas dados pré-inseridos na tabela *Energias* da base de dados pois, à data da execução do protótipo, não tinha sido estabelecida ainda a ligação à base de dados da EDP Distribuição.

O aperfeiçoamento deste bloco passa pela concretização da ligação efetiva à base de dados da EDP Distribuição de forma a exibir os valores pretendidos. Uma vez conhecidos e guardados os valores na tabela da base de dados respetiva, será apresentado um diagrama de carga diário.

Bloco Agenda:

Este bloco exibe um *iframe* na forma de calendário disponibilizado pela “Agenda 7 Coimbra”. Esta plataforma surge de parceria entre a Câmara Municipal de Coimbra e a Universidade de Coimbra e dedica-se a promover os eventos culturais da cidade, tais como Exposições, teatro, música, cinema, dança, desporto, livros e eventos científicos.

Por ser a melhor fonte de informação cultural da cidade e por ter atualização diária além de se tratar de um projeto local, não carece de ser substituída. No entanto, para melhorar a experiência do utilizador do Dashboard, existe a ambição de incluir um *mouseover* no calendário que, à semelhança do próprio calendário do *website* “Agenda 7 Coimbra”, mostre os eventos correspondentes ao dia selecionado pelo utilizador.



Figura 4 - *Mouseover* sobre a agenda do “Agenda 7 Coimbra”

Bloco Qualidade do Ar:

Os valores mostrados neste bloco são retirados do *website* da Agência Portuguesa do Ambiente, APA, (<http://qualar.apambiente.pt/index.php?page=2>). No entanto, à semelhança do bloco Energia Elétrica Consumida, não existe ainda uma recolha de dados em tempo real.

Existe portanto a necessidade de estabelecer um ligação à base de dados da APA, com colaboração da Comissão de Coordenação e Desenvolvimento Regional do Centro (CCDRC), para poder mostrar não só os valores das concentrações de poluentes e/ou constituintes da atmosfera, mas também o índice de qualidade de ar associada.

Bloco Locais de Interesse:

Este bloco mostra sob a forma de pontos assinalados no Google Maps, os locais de maior interesse turístico segundo os *websites* “Turismo de Coimbra” e “Turismo do Centro”. Os locais de interesse reunidos não necessitam de sofrer alterações. No entanto, para enriquecer ainda mais a experiência do utilizador do Dashboard, serão assinalados outros pontos no mapa correspondentes às estações ou locais onde são recolhidos os dados usados nos diferentes blocos do Dashboard.

A ideia é acrescentar camadas ao mapa existente, cada uma a identificar as estações que recolhem valores usados no mesmo bloco: para além da camada locais de interesse, acrescentar uma camada que identifique no mapa as subestações que fornecem a energia elétrica à cidade, outra camada para as estações de monitorização da qualidade do ar, e assim por diante, consoante a especificidade de cada medição.

Bloco Programador de Percursos:

O conteúdo deste bloco pode ser descrito como uma pesquisa de itinerários fornecida pelos SMTUC, que redireciona o utilizador para o seu *website* http://www.smtuc.pt/geral/index_itinerarios.php. Aí são mostrados o trajeto, horários e identificação do transporte responsável pelo mesmo. Sendo a empresa SMTUC fornecedora do

principal meio de transporte público urbano e estando o *iframe* fornecido pela mesma a funcionar normalmente, não existe necessidade de fazer alterações ou substituições. Mas, existindo outros meios de transporte público em Coimbra ou usados para chegar à mesma, faz sentido incluí-los neste bloco.

As carreiras da TRANSDEV fazem a ligação de Coimbra a outros concelhos e distritos espalhados pelo país e são usadas por muitos estudantes e trabalhadores da cidade para se deslocarem de e para casa. Já os comboios da CP são um dos meios principais usados para chegar a Coimbra com origem em pontos mais distantes. A inclusão de uma pesquisa de horários/itinerários destes dois meios de transporte faz sentido na medida em que se pretende que o Dashboard seja uma plataforma de informação tanto para os cidadãos como para os visitantes.

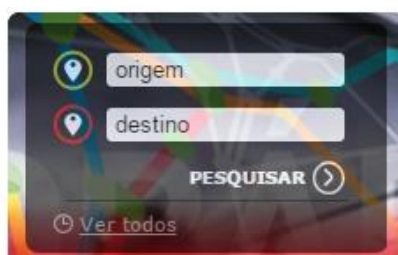


Figura 5 - Pesquisador de itinerários da TRANSDEV

Figura 6 - Pesquisador de itinerários da CP

Bloco Emissões de CO₂:

Neste bloco são determinadas as emissões de dióxido de carbono associadas à produção de Energia Elétrica, exibida no bloco Energia Elétrica Consumida. Embora estes valores sejam aproximados, tendo em linha de conta a importância de estimular o uso sustentável dos recursos energéticos, essa informação irá manter-se inalterada no Dashboard. Todavia, devem ser atualizados anualmente os valores usados para este cálculo, retirados do documento Dados Técnicos, publicado pela REN, e “Recomendação sobre Rotulagem de Energia Elétrica”, disponibilizado pela ERSE.

Uma vez que irão ser acrescentados outros blocos a esta plataforma, entre eles um dedicado ao consumo de água na cidade, é pertinente incluir no bloco emissões de CO₂ as emissões de dióxido de carbono correspondentes à energia elétrica usada no abastecimento e bombagem de água.

2.2 Blocos a acrescentar

Na identificação de potenciais blocos de informação relevante para acrescentar ao conteúdo do Dashboard foram tidas em conta não só as características próprias da cidade como também as sugestões deixadas pelo anterior responsável do projeto para inclusão posterior. Cada uma dessas informações só foi considerada viável caso exista previamente uma fonte ou interlocutor que permita a recolha automática e em formato digital dessa mesma informação.

Na escolha das fontes de informação é dada prioridade a produtores de informação locais mas também são tidos em consideração fatores como a facilidade de obtenção de dados, a frequência com que são adquiridos/atualizados, potenciais custos associados e a disponibilidade dos mesmos em fornecê-la.

Começando pelas sugestões deixadas previamente temos o consumo de água e gás. As restantes informações consideradas para inclusão no Dashboard correspondem ao nível do rio Mondego, quantidade de resíduos recolhidos, farmácias de serviço e eventos desportivos. De seguida são explicadas em detalhe as razões da escolha de cada informação, o que se pretende exibir nos blocos e possíveis fontes dessa informação.

Bloco Consumo de Água:

Este bloco, à semelhança do bloco Energia Elétrica Consumida, pretende disponibilizar os valores dos consumos de água dentro da cidade. Este é um tipo de informação relevante para os cidadãos e exibida em alguns dos dashboards mencionados na secção anterior. À imagem da Energia Elétrica, pretende-se obter valores com a menor periodicidade possível para ir ao encontro da característica dinâmica do Dashboard. Para além do volume de água consumido, pretende-se obter o valor de energia elétrica associada a esse consumo, usada na bombagem da água para todos os pontos da cidade. A empresa a contactar será a Águas do Mondego.

Bloco Consumo de Gás:

Após a inclusão dos consumos de água e eletricidade no Dashboard, a introdução de um bloco destinado ao consumo de gás é tanto óbvia como lógica: pretende-se exibir os consumos de gás dentro da cidade com a máxima frequência possível. A Lusitaniagás é a empresa concessionária para a distribuição de gás na região Litoral Centro, do qual faz parte a cidade de Coimbra, pelo que essa empresa deverá ser a contactada para obter os valores de consumo de gás.

Bloco Consumo de Recursos:

De forma a interpretar os diagramas de consumo de energia elétrica, água e gás considerou-se importante acrescentar um bloco que faça a análise dos dados disponíveis e que mostre indicadores baseados nesses dados. Pretende-se que a inclusão destes indicadores melhore a

experiência de navegação dos utilizadores do dashboard, mostrando informações que de outra forma seriam ignoradas.

Bloco Nível do Rio Mondego:

Tendo sido a cidade de Coimbra construída à volta do Rio Mondego, este figura como presença principal da paisagem urbana. Embora o caudal do rio Mondego seja controlado a montante pela barragem da Aguieira, a pequena flutuação do nível diário do mesmo não deixa de ser um dado interessante para o utilizador do Dashboard, e ainda mais em períodos de seca ou de cheias.

Os caudais e correntes das bacias hidrográficas de norte a sul do país são monitorizados pela APA e exibidos ao público pelo Sistema Nacional de Informação de Recursos Hídricos (SNIRH) através de dados recolhidos em diversas estações colocadas a poucos quilómetros entre si. Em Coimbra, quase todas as pontes têm uma dessas estações, o que permite obter valores rigorosos e precisos do nível do rio à porta da cidade. A SNIRH é então a entidade a contactar de forma a obter tanto o caudal como o nível do rio Mondego.

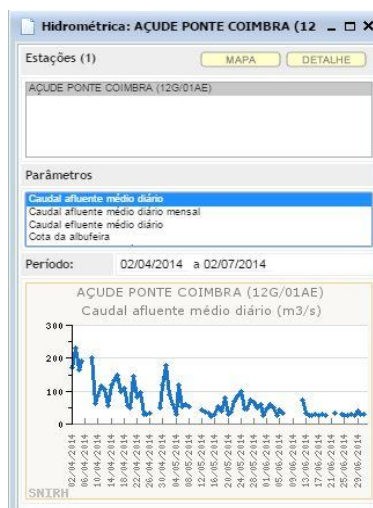


Figura 7 - Caudal afluente médio diário da estação Açude Ponte

Bloco Resíduos:

Com o aumento da população nos meios urbanos e consequente aumento do consumo é cada vez mais importante monitorizar não só os resíduos recolhidos mas também o seu destino. A quantidade de resíduos produzidos diariamente na cidade de Coimbra é um indicador dos hábitos de consumo e a presença desse número no Dashboard servirá para alertar os cidadãos para a importância de reduzir o desperdício e aumentar a reciclagem.

Os resíduos do concelho de Coimbra são recolhidos pela Câmara Municipal de Coimbra e pela empresa ERSUC – Resíduos Sólidos do Centro, S.A.. Estas serão portanto as entidades a contactar para obter os valores pretendidos.

Bloco Farmácias:

Dada a natureza de rotatividade das farmácias de serviço e a importância das mesmas para a saúde dos cidadãos, é eminentemente útil dispor de uma plataforma *online* de consulta rápida do horário e localização das mesmas. O *website* Farmácias de Serviço (<http://www.farmaciasdeservico.net/localidade/coimbra/coimbra>) faz isso mesmo ao disponibilizar aos cibernautas, por cada concelho, as farmácias em funcionamento atualizadas a cada 24 horas. Adicionalmente fornece a localização, em mapa e por endereço, de cada uma e o *hyperlink* quando esta dispõe de presença *web*.

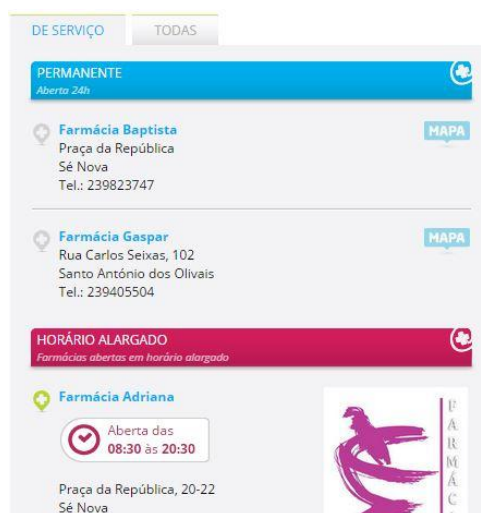


Figura 8 – Parte da informação disponibilizada no *website* Farmácias de Serviço para Coimbra

3. Identificação dos produtores de informação e especificação dos conteúdos

Uma vez conhecidas todas as categorias de informação e blocos que irão figurar no Dashboard final, é preciso enumerar os interlocutores/produtores dessas informações e o tipo de contacto com cada uma delas. Para cada um desses blocos será descrito o método de cedência da informação, qual a informação cedida e a sua forma, a sua frequência de atualização e finalmente a forma como serão exibidas no Dashboard.

A maioria dos contactos com os interlocutores foi feita por correio eletrónico, telefone ou encontro presencial. No final deste projeto contamos com mais de cem trocas de correio eletrónico, dezenas de telefonemas e cerca de duas dezenas de reuniões, quer nas instalações dos interlocutores quer no próprio Departamento de Engenharia Eletrotécnica e de Computadores. Para evitar imprecisões nas referências às posições institucionais dos interlocutores das empresas e entidades que colaboraram no desenvolvimento do projeto, optou-se por omitir estas referências, aludindo-se às pessoas apenas pelos respetivos nomes.

No contacto com os interlocutores foi proposta por parte de UC a celebração de um protocolo, com o respetivo Anexo técnico, com direitos e deveres a serem cumpridos por ambas as entidades envolvidas. No anexo A é possível consultar um Protocolo genérico.

3.1 Meteorologia

Os valores de meteorologia pretendidos são medidos na estação meteorológica do DEM e o seu responsável de momento é o Eng.º João Carrilho. O contacto foi estabelecido primeiro por correio eletrónico e mais tarde por encontro presencial dentro do DEM.

Inicialmente propôs-se usar um dos *iframes* criados automaticamente pelo *website* WeatherUnderground que, como já foi referido, recebe os valores de meteorologia lidos por essa estação. No entanto, devido ao reduzido número de dados existentes nesses *iframes*, iniciou-se uma negociação para criar um *feed* único para este projeto. A estação é do tipo *Davis VantagePro2Plus* e os dados recolhidos são geridos através do *software weewx*. Uma vez que esse *software* facilita a criação de *feed* XML, durante um dos encontros presenciais foram escolhidas as medições da estação que necessitávamos no projeto Dashboard e criado um *feed*. Este *feed*, http://www2.dem.uc.pt/meteo/rss/weewx_rss.xml, atualizado a cada 5 minutos, contém os seguintes índices e valores meteorológicos: temperatura em graus Celcius, velocidade

e direção do vento em metros por segundo (m/s) e graus respetivamente, a pluviosidade em milímetros por hora (mm/hora), pressão atmosférica em hectopascal (hPa), a humidade relativa em percentagem, a radiação solar em Watt por metro quadrado (W/m²) e o índice de UV. Existem outros três valores enviados que, por opção, acabaram por não ser exibidos neste painel: temperatura do vento, índice de calor e ponto de orvalho. O excerto seguinte representa parte do *feed* disponibilizado:

```
<![CDATA[<p> Time: 29-Jul-2015 15:05<br/> Outside Temperature: 28.7°C<br/>
Wind Chill: 28.7°C<br/> Heat Index: 29.2°C<br/> Dewpoint: 17.2°C<br/>
Humidity: 50%<br/> Barometer: 1015.3 hPa<br/> Wind Speed: 3.6 m/s<br/> Wind Direction:
315°<br/> Rain Rate: 0.0 mm/hr<br/> Solar Radiation: 857 W/m<sup>2</sup><br/> UV index:
5.8<br/> </p>]]>
```

Infelizmente esta estação não tem a capacidade de usar as suas medições para determinar o estado do tempo atual. Durante algum tempo considerou-se qual a melhor opção para colmatar esta falha até que foi tomada a decisão de usar um segundo *feed* de meteorologia. O *feed* criado pelo *website* OpenWeatherMap para a cidade de Coimbra, <http://api.openweathermap.org>, que contém para além da informação do estado do tempo atual, outras duas informações sobre as quais o seu uso será descrito na próxima secção: hora do nascer e pôr-do-sol.

No final, as informações que serão exibidas pelo bloco Meteorologia são, por ordem, as seguintes: temperatura, humidade, índice UV, velocidade do vento, direção do vento, radiação do Sol, pressão atmosférica, pluviosidade e estado do tempo atual.

3.2 Nível do Rio

O contacto com o SNIRH foi feito maioritariamente por correio eletrónico e mais tarde por telefone. Este contacto foi feito diretamente com a Dra. Manuela Saramago e Eng^a. Cláudia Brandão que mostraram desde início todo o apoio a este projeto. Embora o nível hidrométrico do Rio Mondego seja medido em várias estações perto da cidade de Coimbra, apenas a estação Ponte de Santa Clara e estação Açude Ponte estão de momento a funcionar. Para acedermos a esses valores foi-nos criada uma conta de acesso no portal SNIRH onde é possível consultar uma tabela com as medições, hora a hora, de cada uma dessas estações. No entanto, este acesso não permite obter esses valores de forma automática pelo que foi sugerido da parte da Dra. Manuela Saramago e Eng. Cláudia Brandão, o envio dos valores por FTP (*File Transfer Protocol*).

Até ao momento da conclusão desta versão do Dashboard este envio ainda não tinha sido programado, no entanto avista-se uma resolução rápida. Todavia, até esse envio estar completo estão a ser usados valores aleatórios no bloco Nível do Rio.

3.3 Qualidade do Ar

O Primeiro contacto com a APA foi feito com a Dra. Ana Sousa, e embora tenha sido mostrado desde o início a sua vontade de colaborar neste projeto, as limitações técnicas dificultavam o acesso aos valores de concentração de gases poluentes quer à base de dados da APA, quer às próprias estações de medição. Surgiu mais tarde da parte do Eng. Jorge Landeck¹, durante uma reunião, a solução para obtenção dos valores, após ser esclarecida detalhadamente a forma como são obtidos estes valores por parte da APA.

Em cada estação existe um *modem* analógico que envia, em resposta a uma interrogação, os valores medidos continuamente para o servidor central da CCDRC, onde são guardados numa base de dados. Estas medições são posteriormente enviadas, com recurso ao sistema Atmis, para o servidor da APA a cada hora. Na APA os valores são revistos e caso se encontrem anomalias, estas são reportadas à CCDRC que confirma os valores ou envia novas medições. Os valores já validados são depois usados no sistema Qualar na APA para determinar o índice de qualidade do ar.

A solução apontada pelo Eng.º Jorge Landeck, passa por replicar o envio de dados para a APA diretamente para o nosso servidor. Este envio é feito pelo método GET e a comando de entrega dos valores tem o seguinte aspeto:

```
http://193.136.121.109/inserir.php?login=ola&passwd=ola&accas=poluente&id_estacao=2016&dia_ref_fim=13&mes_ref_fim=12&ano_ref_fim=2005&hora_fim=16:00&co8h=&no2=18.623&o3=41.000&pm10=21.645&so2=5.320&pm25=&benz=&status=1&dia_ult_val=13&mes_ult_val=12&ano_ult_val=2005&hora_ult_val=18:33
```

O comando envia várias informações, entre as quais as mais importantes são as médias horárias das concentrações de Ozono, Dióxido de Azoto, Monóxido de Carbono, Dióxido de Enxofre e as partículas inaláveis ou finas, cujo diâmetro médio é inferior a 10 micra* (PM10). No entanto, o índice de qualidade do ar, por ser apenas calculado do lado da APA, não é enviado.

Após várias reuniões no CCDRC com a Eng^a. Cristina Taliscas e Eng^a. Helena Brandão, foi programado o novo envio e foi dado auxílio no método do cálculo do índice de qualidade do ar. Este cálculo é explicado detalhadamente no Apêndice B.

3.4 Energia Elétrica

No início da criação da nova versão do Dashboard era possível consultar os dados relativos às três subestações que abastecem a cidade de Coimbra. O acesso a estes valores *online* foi cedido pela EDP Distribuição por intermédio do Eng.º António Amorim, que criou uma conta de cliente virtual permitindo a consulta dos valores de consumo.

¹ Responsável pelo desenvolvimento do sistema Atmis.

Sendo este o único acesso disponibilizado pela EDP, desde o primeiro momento que se tentou aceder a estes mesmos valores de forma automática, simulando em código o acesso à camada final onde se encontram as tabelas, mas sem sucesso. A primeira tentativa foi feita com o cliente HTTP Postman e posteriormente com o *software* cURL.

Uma vez que estes instrumentos não nos permitem um acesso automático aos valores de consumo de energia elétrica, tentou-se insistir com outro tipo de processo de recolha de dados. Até à data da conclusão desta versão ainda não tinha sido estabelecido o acesso pretendido e como tal, para testar as funcionalidades dos diagramas de carga dinâmicos que foram criados para o Dashboard, foram usados os valores de consumo de energia elétrica do DEEC. Estes valores, atualizados com uma frequência de 15 minutos, são cedidos pela empresa Streamline, e estão alojados na base de dados *mewago*.

O diagrama de carga da energia elétrica é composto por duas curvas, uma para o consumo do dia atual e outra para dia anterior. As razões para o uso de uma segunda curva são para permitir uma comparação direta do consumo do dia atual com o anterior mas também para que nas primeiras horas de um novo dia o diagrama não aparentar estar vazio.

3.5 Água

A gestão do abastecimento de água da região Centro é feita pela Águas do Mondego, geridas pelo Eng.º Nelson Geada, e foi com que ele que foram feitos os primeiros contactos. Após uma reunião nas instalações das AM, foi o Eng.º Nelson Moura que ficou responsável pelo auxílio a este projeto. Durante as reuniões, tornaram-se aparentes as dificuldades técnicas em permitir o acesso ou até do envio dos valores de consumo de água e da respetivo gastos energéticos associados por parte do seu sistema de controlo de dados. A solução sugerida foi então de replicar, em cada ponto de entrega de água, num total de 5, a medição do caudal.

Até ao momento da conclusão desta versão do Dashboard ainda não havia um acordo quanto à melhor forma de obter os dados, por esse motivo, à semelhança dos valores de energia elétrica, foram usados os valores de consumo de água do DEEC, cedidos pela empresa Streamline, através da base de dados *mewago*.

Também à semelhança do gráfico de energia elétrica, este gráfico também é composto por duas curvas, uma do consumo de água do dia atual e outra do dia anterior, pelas mesmas razões apresentadas anteriormente.

3.6 Gás

O primeiro contacto com a empresa LusitâniaGás foi feito com o Eng.º Gabriel Charrua de Sousa, por contacto eletrónico e, mais tarde, por encontro presencial. Nessa reunião, foi esclarecido que o volume de gás consumido, em metros cúbicos, não é monitorizado com elevada frequência sendo apenas medido uma vez por dia. Além disso, para uma contagem de consumo que englobe a cidade e arredores é necessário usar 6 pontos de entrega, são eles: Condeixa, Cernache, Casa do Sal, Oliveira do Hospital, Loreto, e Santa Clara.

A forma de cedência desses valores indicada pela empresa é a de envio por FTP de um ficheiro Excel, que é atualizado diariamente entre as 7h00 e as 11h00 com os valores de consumo de gás em todos os pontos mencionados, entre as 00h00 e as 23h59 do dia anterior. Este ficheiro Excel corresponde apenas ao consumo de um mês, sendo que todos os meses é criado um novo ficheiro. O nome de cada ficheiro engloba a data de início e fim do mês em questão, como por exemplo “LTG_PRM_01062015_30062015”, ficheiro que corresponde ao mês de Julho. Cada um desses ficheiros está dividido em separadores, cada um deles com o nome de uma das PRM (Ponto de Regulação e Medição), e com uma tabela à qual é acrescentada todos os dias uma nova linha com o consumo do dia anterior.

A representação gráfica do consumo do gás será feita num intervalo de tempo semanal devido à reduzida frequência de cedência de dados (diária). Mais uma vez optou-se por usar duas curvas na representação gráfica, uma para o consumo de gás da semana atual e outra para a semana anterior. As razões desta escolha são as mesmas apresentadas nas secções anteriores.

3.7 Consumo de Recursos

O “consumo de recursos” consiste na apresentação de alguns indicadores, resultantes de uma análise ao consumo de todos os recursos reunidos e exibidos no Dashboard: energia elétrica, água e gás. Esta análise destina-se a permitir ao visitante do Dashboard uma interpretação mais rica e informada sobre o desempenho da cidade na utilização de recursos, utilizando indicadores calculados com base nos valores das variáveis de consumo recolhidos. Os dados sobre estes indicadores são apresentados num painel autónomo, importando neste passo identificá-los.

No caso do consumo de energia elétrica optou-se por mostrar indicadores relativos apenas ao dia atual, tais como os valores máximos e mínimos, potência média e fator de carga. A potência média é definida como o quociente entre a energia gasta num período de tempo e a duração desse mesmo período. Já o fator de carga, um índice que ilustra a eficiência da utilização da infraestrutura elétrica, é obtido pela relação entre a potência média e a potência máxima, durante

um período de tempo. Esse período de tempo corresponde ao tempo decorrido até ao momento do dia atual.

No caso do consumo de Água escolheu-se mostrar para além dos valores máximos e mínimos de consumo, o consumo médio do dia atual até ao momento e no período homólogo do dia anterior. É também feita uma comparação do consumo do dia atual com o necessário para encher uma piscina com 80 m³ de capacidade (o valor reduzido de capacidade é usado uma vez que apenas temos acesso ao consumo de um departamento).

Finalmente no caso do consumo de gás é mostrado o valor máximo da semana atual e da semana anterior, o valor médio da semana atual e do período homólogo da semana anterior. É feito também a conversão do consumo de gás do dia anterior, em metros cúbicos, para MW/h. Esta conversão é feita segundo o valor de conversão da Agência Internacional de Energia[11], segundo o qual 93 m³ de gás natural correspondem a 1 MW/h.

3.8 Produção de Resíduos

O primeiro contacto para obter os valores de produção de resíduos foi com a Câmara Municipal, que rapidamente indicou a Divisão do Ambiente do Departamento de Desenvolvimento Social e Ambiente como responsável pelo tratamento desses valores. Após várias trocas de *email* com a Divisão, reunimos nas suas instalações com a Eng^a. Inês Carvalho que explicou o processo de recolha, pesagem e tratamento dos resíduos e destino final de cada tipo. Ficou evidente que os valores de pesagem de resíduos não poderiam ser fornecidos pela Divisão, mas sim diretamente com a ERSUC. Ainda assim, forneceram os valores de resíduos produzidos e tratados no concelho de Coimbra nos últimos três anos disponíveis: 2011, 2012 e 2013. Esta informação foi enviada num ficheiro Word com os valores em toneladas discriminados por diversos tipos: Resíduos Urbanos Recolhidos de forma indiferenciada; Resíduos de embalagens recolhidos em ecopontos; Resíduos de embalagens recolhidos fora dos ecopontos; Bio-resíduos; REEE; Oléos alimentares recolhidos; Pilhas e acumuladores; Volumosos; Têxteis; Resíduos urbanos indiferenciados recolhidos em grandes produtores; Outros resíduos urbanos recolhidos. Para além destas informações, para cada ano foram também fornecidas informações relativas aos equipamentos de deposição indiferenciada e seletiva de resíduos e às emissões gasosas produzidas pelos veículos de recolha.

Numa tentativa de ter acesso sistemático e automático às quantidades de resíduos recolhidos em Coimbra contactou-se o Eng.º João Braga da empresa ERSUC. Numa reunião presencial nas instalações da ERSUC discutimos a possibilidade de recebermos esses valores, ou de interrogar o sistema responsável pelo seu armazenamento, à maior frequência possível mas não houve

abertura da parte da empresa para esse afeito. Até à data da conclusão desta versão do Dashboard não houve uma resposta concreta a esse pedido.

3.9 Mobilidade

O objetivo desta nova versão do Dashboard em relação ao bloco mobilidade é de acrescentar diferentes consultas de meios de transportes em Coimbra, ao já existente, os SMTUC. Para tal foram enviados *emails* tanto para a empresa TRANSDEV como para a CP a pedir a disponibilização de um *iframe* que pudesse ser incorporado no Dashboard. Da parte da CP não houve resposta às várias tentativas de contacto, enquanto da TRANSDEV chegou a existir uma breve troca de correio eletrónico onde a empresa mencionou a possibilidade de disponibilizar um *iframe* no final do primeiro trimestre do ano. Desde essa altura foram feitas novas tentativas de contacto todas sem sucesso. Ficou então aparente que a única empresa de transportes que iria figurar no Dashboard seria a SMTUC.

Pretendia-se também fazer um pequeno ajuste ao *iframe* da SMTUC, uma vez que neste momento, ao fazer uma pesquisa ou clicar em “Pesquisa Avançada” o *website* da SMTUC é aberto no mesmo separador do Dashboard em vez de abrir num novo separador. Foram feitas várias tentativas de contacto com a SMTUC para tentar solucionar esta situação mas nenhuma obteve resposta.

4. Escolha das ferramentas de suporte e construção do Dashboard

Após escolhidos todos os tipos de dados que irão figurar no Dashboard final o passo seguinte é a sua construção. Em primeiro lugar foi preciso optar entre continuar o trabalho desenvolvido na primeira versão do protótipo do Dashboard ou fazer uma nova abordagem ao projeto. A primeira opção foi descartada, tendo sido decidido iniciar uma nova versão do Dashboard mais funcional e com um visual mais apelativo, usando um conjunto de novas ferramentas que permitiram um desenvolvimento mais versátil.

Para agilizar o desenvolvimento do novo protótipo começou-se por instalar um servidor local, o XAMPP. O XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo HTML com suporte à linguagem PHP, fundamental para o nosso projeto.

A estruturação da interface da nova versão foi baseada no *framework* Twitter Bootstrap[11]. O Bootstrap, como é normalmente conhecido, é uma coleção de ferramentas e bibliotecas viradas para a criação de *websites* e aplicações *web* utilizando HTML e CSS. Além disso, o Bootstrap possui ainda componentes ou *plugins* em JavaScript e CSS que permitem dinamizar os componentes da página *web*. Com estas ferramentas foi possível desenvolver a base da nova versão do Dashboard, desde a organização da página ao *design* da mesma.

Tal como no protótipo anterior, foi usado o suporte CakePHP, também ele usado no projeto principal, o Smart City Coimbra. O CakePHP é um *framework* escrito em PHP que permite desenvolver aplicações robustas e bem estruturadas sem perder a flexibilidade. Este *framework* usa a arquitetura MVC (*Model, View, Controller*, ou Modelo, Vistas, Controlador).

Após a instalação do CakePHP na máquina usada para desenvolvimento, foi necessário fazer a integração do Bootstrap de modo a viabilizar a utilização da interface previamente criada. Este processo é simples e consiste na cópia de alguns ficheiros e pastas para diretorias específicas do CakePHP.

Para desenvolver o código, criar e alterar ficheiros do CakePHP recorreu-se ao NetBeans IDE, um ambiente de desenvolvimento integrado gratuito e de código aberto, que permite desenvolver projetos nas linguagens JavaScript, PHP entre outras.

Uma vez preparada a plataforma que irá mostrar os diversos valores, faltava ainda construir uma base de dados para os armazenar. Para tal foi usado o *software* MySQL Workbench versão

6.2 para aceder ao servidor e foi criada uma base de dados denominada “dashboard” com as tabelas adequadas às necessidades do protótipo.

Nesta fase estavam reunidas todas as condições para o desenvolvimento normal do projeto e estavam a ser usadas as seguintes linguagens: CSS, HTML, PHP, SQL e JavaScript. Com o desenrolar do projeto, foi necessário procurar uma forma de atualizar os valores das variáveis que requerem frequência alta de refrescamento sem recorrer ao recarregamento da página. Para este efeito escolheu-se um *framework* denominado AngularJs, que embora seja um *framework* JavaScript, adota uma abordagem mais ligada à sintaxe HTML, funcionando como uma espécie de extensão da linguagem, e que segue também o padrão MVC. O AngularJs é usado nesta versão do Dashboard em todos os valores dinâmicos, ou seja, todos os valores que sofram alterações a uma frequência elevada.

Outra ferramenta usada no desenvolvimento de respostas dinâmicas no painel foi o JQuery, uma biblioteca JavaScript para manipulação do DOM, *Document Object Model*, (ex. mudança de cores de um elemento, eventos *onclick*, animação). Na execução do JQuery, o seu código faz chamadas a funções de uma biblioteca, solicitado por algum evento DOM.

Para a conceção dos vários gráficos a escolha recaiu sobre o “Highcharts”. O Highcharts é uma biblioteca de gráficos escritos inteiramente em JavaScript, que oferecem uma forma fácil de adicionar gráficos interativos a uma página *web*. Na sua biblioteca estão disponíveis os mais variados tipos de gráficos, tais como área, linha, colunas, barras, entre outros. As grandes vantagens do Highcharts são: ser suportado em todos os navegadores modernos, não necessitar de *plugins* do lado do cliente, tais como Flash ou Java, e ter uma sintaxe de configuração simples.

Ao longo do processo de criação do painel Dashboard, o seu desenvolvimento foi feito localmente sendo, posteriormente a cada alteração, feito o *upload* para o servidor que o irá alojar. Para esse efeito foi usado o sistema de controlo de versão Git. O Git permite criar um diretório de trabalho em que cada repositório tem o histórico completo do projeto, neste caso, todos os estados de evolução do projeto. O interface *web* usado para comunicar com o Git neste projeto é o GitLab e para executar os comandos do Git foram usados a *shell* Git Bash e o *software* PuTTY.

Alguns ficheiros criados durante o desenvolvimento requerem ser executados automaticamente a uma frequência pré-definida. Dado que o servidor onde o projeto está instalado tem como sistema operativo o Linux, foram usados *cron jobs* para executarem os ficheiros com determinados intervalos. *Cron Job* é uma tarefa agendada executada ao ritmo a que é programada.

Após a descrição das principais ferramentas, *software* e linguagens usados na conceção do Dashboard, é preciso especificar o uso de cada uma delas nas várias fases do seu desenvolvimento. Enquanto algumas destas ferramentas são fundamentais e do seu uso depende

a própria estrutura do projeto, outras há que são específicas a pequenas partes do projeto. Seguindo uma lógica de hierarquia vamos começar por detalhar as ferramentas usadas globalmente e posteriormente iremos descrever, painel a painel, todos os seus componentes específicos.

4.1 Bootstrap

O primeiro passo deste projeto foi o desenvolvimento do *layout* base para acomodar o Dashboard, com recurso ao *framework* Twitter Bootstrap. A instalação deste consiste apenas no *download* da estrutura zipada [12].

Para construir uma página *web* com este aplicativo basta escolher um *template* HTML ou um dos muitos exemplos provenientes do próprio Bootstrap e personalizá-lo, adaptando-o às necessidades da página pretendida. Uma vez escolhidos os componentes que irão figurar e a sua posição, a aparência é personalizada nos ficheiros `bootstrap.min.css`, `bootstrap.min.js` e `bootstrap-theme.min.css`.

Usando o London Dashboard como referência, foi criado um painel simples e sóbrio, dividido em três secções principais: cabeçalho, conteúdo e rodapé.

O cabeçalho é composto pelo logotipo do Smart City Coimbra, e pelo título, “SmartCity Coimbra” à esquerda e pela data e hora à direita.

O conteúdo da página, à semelhança do Dashboard de Londres, é subdividido em pequenos painéis, cada um destinado a exibir uma categoria de valores distinta. A disposição dos painéis foi escolhida com base não só na importância e frequência de atualização, como também pelo espaço que cada um iria ocupar, uma vez preenchidos com os respetivos conteúdos.

Finalmente o rodapé, à semelhança do rodapé construído para o protótipo anterior, consiste numa barra com um texto clicável, “Acerca”, que abre uma página (*pop up*) com informações sobre o projeto SmartCity Coimbra Dashboard.

A paleta de cores escolhida é baseada no logotipo já existente, do SmartCity Coimbra, desenhado a preto e branco. Para o conteúdo escolheram-se diferentes variações de cinzento, sendo a cor do texto preta. A única exceção é a hora e data que, para se diferenciar do título, foi escrita num tom dourado escuro. O tipo de letra e tamanho, dimensões da página e dos seus conteúdos, foram sofrendo alterações ao longo do processo de desenvolvimento.

O produto resultante desta fase é mostrado na figura seguinte, onde se pode observar o esqueleto do que irá ser o Dashboard de Coimbra.

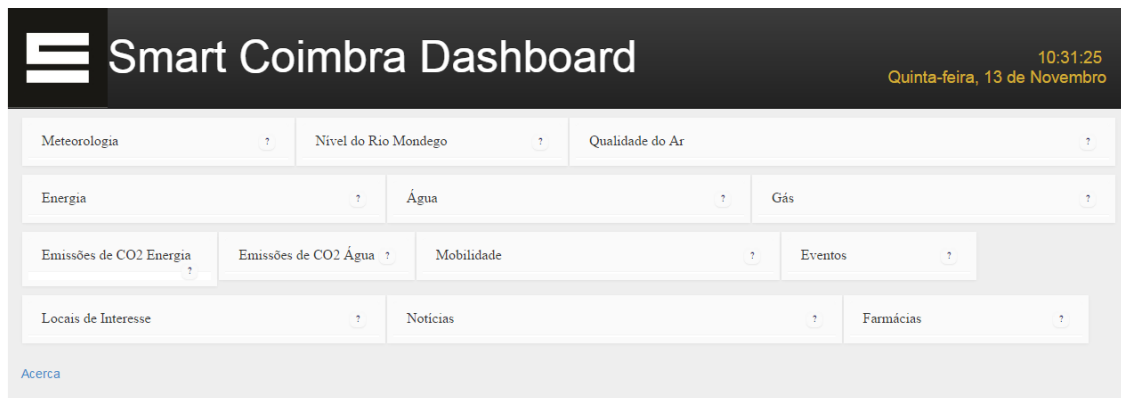


Figura 9 - Fase inicial de construção

4.2 Base de Dados

A primeira base de dados criada, com recurso ao MySQL Workbench 6.2, foi denominada “dashboard” e estava alojada no computador local, no servidor MySQL do XAMPP. Numa fase mais avançada do projeto, foi feita a migração desta base de dados para uma que seria a definitiva, denominada também por “dashboard”, num servidor do domínio deec.uc.pt. Nesta base de dados foram criadas 6 tabelas, em regra uma por cada painel, com as seguintes nomenclaturas: *meteorologias*, *qualidades*, *qtrafechos*, *rios*, *gases* e *qualidades_medias*.

A ligação da base de dados ao projeto em si é feita com um script denominado “database.php” que se encontra na diretoria Source Files/ app/ Config do CakePHP. Na transcrição seguinte encontra-se o código que faz essa ligação à base de dados (com nomes fictícios), dando a essa ligação o nome de “default”.

```
public $default = array(
    'datasource' => 'Database/Mysql',
    'persistent' => false,
    'host' => 'db.deec.uc.pt',
    'login' => 'login_name',
    'password' => 'password',
    'database' => 'dashboard',
);
```

Devido ao atraso no acesso a alguns dos valores mais pertinentes do Dashboard durante o seu desenvolvimento, nomeadamente de energia elétrica e água, recorreu-se a outra base de dados denominada “mewago”, para ir testando e validando as funcionalidades em desenvolvimento. Esta base de dados está ligada ao servidor mewago.streamline.pt e nela são guardados, com intervalos de quinze minutos, os valores de potência de consumo de energia elétrica e de consumo de água do Departamento de Engenharia Eletrotécnica e de Computadores. O acesso a esta base de dados é feito de forma semelhante à anterior, desta vez com o nome de DEEC. Esta BD tem várias tabelas, mas apenas uma tabela, a *Registers*, foi usada. Nesta tabela estão alojados os valores de potência elétrica consumida e de água consumida.

A forma como os diferentes valores são guardados varia de fonte para fonte, mas a maioria depende de um *script* escrito em PHP que obtém os valores antes de os guardar na devida tabela. Nas linhas de código seguintes é exemplificado o processo pelo qual os valores de meteorologia são armazenados no ficheiro *meteo.php*. Em primeiro lugar é estabelecida uma ligação à BD e posteriormente, em caso de sucesso, são guardados na tabela *meteorologias* os diferentes valores lidos de um *feed* com o comando “INSERT INTO” na linguagem SQL.

```
...
$conn = new mysqli ($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);}
...
$sql = "INSERT INTO meteorologias (datahora, temp, humidade, estado_tempo, vento, imagem_url, pressao, direcao, chuva,
radiacao, uv) VALUES ('$datahora', '$temperatura', '$humidade', '$estado_do_tempo', '$vento', '$imagem', '$pressao', '$direccao',
'$chuva', '$radiacao', '$UV')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully"; } else {
    echo "Error: " . $sql . "<br>" . $conn->error; }
$conn->close();
```

A leitura dos valores é feita de forma semelhante, quando esta é feita por um script php, mas desta vez com o comando “SELECT FROM”. Na estrutura do CakePHP a obtenção dos valores segue um processo diferente, uma vez que a própria estrutura já está conectada à base de dados. No exemplo seguinte é mostrado como são obtidos os valores da tabela meteorologia dentro do CakePHP.

```
public function latest() {
    return $this->Meteorologia->find('first', array('order' => array('id' => 'DESC')));
}
```

Na diretoria /Source Files/app/Controller foi criada no ficheiro MeteorologiasController.php a função *latest()* que irá seleccionar a última linha da tabela, a mais recente. Este código é feito em PHP e só é possível devido à forma singular da estrutura do CakePHP. No próximo capítulo a estrutura do CakePHP será explicada em pormenor.

4.3 CakePHP

Como foi referido anteriormente, o CakePHP trabalha sobre o modelo MVC e traz uma estrutura bem organizada e de fácil compreensão, além de outras funcionalidades que agilizam o desenvolvimento de aplicações *web*. A arquitetura MVC (Model, View e Controller) é uma abordagem de *software* que separa a lógica da aplicação da sua apresentação. Na prática, ela permite que as aplicações *web* contenham o mínimo de *scripts* já que a apresentação é separada do código [13][14].

A camada Model representa a parte da aplicação que implementa a parte lógica. É responsável por obter os dados, convertendo-os em conceitos significativos para a aplicação, assim como processar, validar, associar e qualquer outra tarefa relativa ao tratamento dos dados.

A camada View é a camada de apresentação, a interface que será mostrada ao utilizador. Sendo separada do objeto Model, é responsável por usar as informações disponibilizadas para produzir qualquer interface de apresentação que a aplicação possa necessitar.

A camada Controller (controlador) recebe as requisições de dados e invoca a camada Model para apresentar uma resposta para a camada View.

As primeiras alterações feitas à estrutura do CakePHP após o seu *download* são, como foi referido nos capítulos anteriores, a integração do Bootstrap e a ligação à base de dados. Este último processo foi explicado ao pormenor na secção anterior, ao passo que a integração do Bootstrap será explicada de seguida.

A integração é simples e consiste em copiar alguns ficheiros para pastas específicas:

- Copiar os ficheiros `bootstrap.min.css` e `bootstrap-theme.min` para `app/ webroot / css`;
- Copiar o ficheiro `bootstrap.min.js` para `app/ webroot/ js`;
- Copiar a pasta “font” para `app/ webroot`.

Esta é a integração genérica do Bootstrap no CakePHP. No entanto, uma vez que são usados *plugins* específicos do Bootstrap na dinamização do Dashboard, é necessário copiar igualmente as suas bibliotecas para a estrutura do CakePHP. Os ficheiros `modal.js`, `tab.js` e `caroussel.js` foram inseridos na pasta `app/ webroot/ js`.

Após esta adaptação é possível copiar todo o código criado anteriormente no Bootstrap para o ficheiro `templateDashboard.ctp` na diretoria `app/ view /layouts`. Este processo, e todas as alterações feitas a partir deste ponto, são feitos com o *software* NetBeans.

Após este passo é necessário alterar o cabeçalho deste ficheiro para incluir todos os ficheiros que foram copiados para a estrutura:

```
<head><?php
    echo $this->fetch('meta');
    echo $this->fetch('css');
    echo $this->fetch('script');
    echo $this->Html->script('clock');
    echo $this->Html->script('jsapi.js');
    echo $this->Html->script('jquery.min.js');
    echo $this->Html->script('modal.js');
    echo $this->Html->script('caroussel.js');
    echo $this->Html->script('tab.js');
    echo $this->Html->script('angular.min.js');
    echo $this->Html->script('angularjs/graficosView/highcharts-ng.js');
    echo $this->Html->script('angularjs/angular-route.js');
    echo $this->Html->charset();
    echo $this->Html->meta('icon');
    echo $this->Html->css('bootstrap.min');
    echo $this->Html->css('bootstrap-theme.min.css');
    App::import('Vendor', 'modal');
?> ... </head>
```

Para além dos ficheiros mencionados, existem outros que também são incluídos no cabeçalho. São eles o `clock.js`, o ficheiro que contém o código que exibe a hora e data no cabeçalho da página, `jsapi.js`, uma biblioteca JavaScript e `jquery.min.js`, um *framework* JavaScript mencionado anteriormente.

Ao longo do desenvolvimento foram criados mais ficheiros, uns em JavaScript outros em PHP, que foram sendo adicionados ao projeto. Entretanto, outros *frameworks* e bibliotecas (AngularJs, Highcharts, etc..) foram acrescentados consoante surgiu a necessidade de os usar em determinados painéis. A sua inclusão será explicitada no momento em que a construção do painel que deles carece for pormenorizada.

Para não sobrecarregar o ficheiro `templateDashboard.ctp`, todos os códigos e funções criadas sobre os valores recolhidos da base de dados foram feitos ou em ficheiros à parte, que seriam depois incluídos (por exemplo o `clock.js`), ou escritos no ficheiro `app/controller/PagesController.php`. Deste modo, `templateDashboard.ctp` contém apenas estruturação do Dashboard em HTML e o que irá ser exibido em cada painel.

O ficheiro `PagesController.php` é um controlador padrão que vem com a estrutura CakePHP e é usado para servir páginas estáticas, neste caso, `templateDashboard.ctp`. Os valores armazenados na base de dados são importados com uma função definida no respetivo controlador, com o mesmo nome da tabela da base de dados que está a aceder (por exemplo, em `MeteorologiasController.php` é definida a função que irá buscar a última linha da tabela *meteorologias* da base de dados). As funções são invocadas em `PagesController.php` e para cada função definida num diferente controlador é necessário o seguinte passo:

```
App::import('Controller', 'Meteorologias');
$meteorologias = new MeteorologiasController;
$out = $meteorologias->latest();
$this->set('meteodata', $out);
$imagem=$out['Meteorologia']['imagem_url'];
$this->set('imagem', $imagem);
```

4.4 AngularJs

O AngularJs é um *framework* estrutural, desenvolvido em 2009 pela equipa da Google, que tem como objetivo simplificar o desenvolvimento de aplicações *web*. É escrito inteiramente em JavaScript e permite estender a sintaxe HTML de forma a exprimir os componentes das aplicações de forma clara e sucinta [15].

O uso desta plataforma é seguramente o componente mais relevante de todo o projeto pois é ele que assegura a distinção entre o *website* em construção, um *website* que aspira ser um Smart City Dashboard, de uma outra qualquer página *web*. O AngularJs permite ao Dashboard fazer uma atualização de dados automática e autónoma, isto é, sem a imposição de fazer o

recarregamento de toda a página para visualizar os valores atualizados. Por este motivo será dado maior destaque a este componente do projeto.

A utilização do AngularJs permite simplificar quer a camada de desenvolvimento, quer obter resultados mais interessantes em termos de *User Experience* sem aumentar exageradamente a complexidade da solução ou os conhecimentos técnicos requeridos. As principais características desta *framework* são o “Data-binding” bidirecional, o padrão de desenvolvimento MVC, a personalização de diretivas, injeção de dependências, comunicação com o servidor entre outras. Os principais conceitos usados neste projeto são descritos de seguida.

4.4.1 MVC

Este padrão de desenvolvimento *web* é, tal como no caso do CakePHP, composto por três partes:

- Model ou Modelo: é o nível mais baixo do padrão e é responsável por gerir os dados;
- View: É responsável por exibir toda ou parte dos dados ao utilizador. A view corresponde ao *template* da aplicação, neste caso o ficheiro `templateDashboard.ctp`;
- Controller ou Controlador: contém o código que controla as iterações entre o model e a view. Os controladores do AngularJs são normalmente mantidos em ficheiros à parte.

4.4.2 Diretivas

As diretivas ou “directives” são um dos conceitos mais poderosos do AngularJs. São elas que permitem estender a linguagem HTML em respostas às necessidades da aplicação. O AngularJs vem com várias diretivas definidas, algumas das quais foram usadas no projeto, tais como “ng-app”, “ng-controller”, “ng-repeat”, “ng-cloak”. As principais no entanto são a “ng-app” e “ng-controller”.

A diretiva “ng-app” é usada para inicializar a aplicação AngularJs e encontra-se no topo da view do projeto:

```
<html ng-app="dashboard">
```

A diretiva “ng-controller” define o controlador. Os controladores permitem, mediante programação, implementar a lógica de apresentação do AngularJs. Servem principalmente para dividir certas partes do código e neste projeto foi criado, no mínimo, um controlador para cada painel que faça uso do AngularJs, num total de 11 controladores. Os controladores são criados em ficheiros à parte, estando a maioria em `app.js`, sendo depois inicializados na view:

```
<div class="Meteo" ng-controller="meteoController">
```

O código acima indica que o controlador “meteoController” deve ser utilizado para controlar o conteúdo do elemento `<div>` onde o mesmo foi declarado.

4.4.3 Data-Binding

O AngularJs tem a capacidade de fazer a ligação direta e bidirecional dos dados (Data Binding bidirecional) que permite a sincronização automática entre models e views. Desta forma é reduzida a quantidade de códigos para mostrar os dados processados pelo servidor. Neste projeto os dados fluem apenas num sentido, do lado do servidor para o lado do cliente. No entanto, esta característica do AngularJs pode ser importante em futuros desenvolvimentos do projeto, no caso de se implementar um painel interativo com o visitante.

4.4.4 Módulos

No AngularJs, um módulo é uma coleção de componentes. Módulos devem ser utilizados para agrupar objetos que contribuem entre si para algum objetivo e também para facilitar a reutilização de comportamentos. Assim, uma aplicação é composta por um ou mais módulos mas neste projeto existe apenas um módulo denominado “modulo” que é definido no ficheiro app.js:

```
var modulo =angular.module('dashboard', ['highcharts-ng']);
```

Na definição do módulo podemos ver dois parâmetros: “dashboard”, o nome dado à aplicação com a diretiva “ng-app” e “highcharts-ng”. Este último é uma diretiva criada especificamente pelo Highchart para que se possam usar os seus gráficos em conformidade com o AngularJs.

4.4.5 Controladores

Com foi descrito na secção 4.4.2, existe neste projeto um controlador por cada painel. São eles denominados “mondegoController”, “meteoController” e “qualidadeController” para os painéis Meteorologia, Nível do Rio e Qualidade do Ar, “CEnergiaController”, “CAguaController” e “CGasController” para o painel Consumo de Recursos, “EmissoesController” para o painel Emissões de CO₂ e finalmente um controlador para cada gráfico: “graficoEnergiaController”, “graficoAguaController” e “graficoGasController”. Os primeiros controladores estão definidos em app.js enquanto os três últimos são definidos cada um num ficheiro com o mesmo nome. Todos estes ficheiros estão na diretoria app/ webroot/ js. Um controlador é assim definido:

```
modulo.controller('meteoController',function($scope,$http,$interval){ ... });
```

4.4.6 Serviços

Um serviço é outro componente do AngularJs que é usado para organizar e partilhar código transversalmente à aplicação. Serviços são funções responsáveis por executarem tarefas específicas, tendo nesta plataforma sido usados 3 serviços: \$scope, \$http e \$interval.

- \$scope

O \$scope é um serviço usado para conectar os controladores às views para as quais foram ligados com a diretiva “ng-controller”. Um controlador pode adicionar valores ou funções ao seu \$scope que depois estarão visíveis na view:

```
modulo.controller('meteoController', function($scope,$http,$interval) {  
    ...  
    $scope.meteo=data;  
    ...  
});
```

No exemplo anterior, na view a variável “meteo” terá o valor que lhe é associado pelo \$scope do controlador meteoController. Quando ocorrem alterações no \$scope, é disparado um evento. Com estes eventos a view sabe quando deve refrescar a variável “meteo”.

- \$http

O \$http é um serviço do AngularJs usado para ler valores de um servidor remoto. Este serviço é fundamental neste projeto pois é com ele que são executados os ficheiros que adquirem os valores pretendidos da base de dados:

```
modulo.controller('meteoController', function($scope,$http,$interval) {  
    var meteodata = function() {  
        $http.get('/projeto/meteo_bd.php')  
            .success(function(data) {  
                $scope.meteo=data;  
            }).error(function(data) {  
                ...  
            });  
    };  
    ...  
});
```

Este serviço usa apenas um parâmetro, o URL ou diretório do ficheiro a que queremos aceder, que é usado para gerar um pedido HTTP, e devolve uma “promise” (ou promessa), com dois métodos específicos: “success” (no caso de sucesso no acesso ao ficheiro) e “error” (no caso da ocorrência de algum erro).

- \$interval

Este serviço do AngularJs é usado para controlar o intervalo de ação de uma determinada função. Os seus parâmetros são a função que está a controlar e o tempo, em milissegundos, entre duas iterações seguidas:

```
var repeatFunction = $interval(meteodata, 60000);
```

É com estes três serviços que é feita a atualização automática dos valores apresentados no Dashboard. O serviço \$http.get lê os valores pretendidos a cada intervalo definido pelo serviço \$interval. Caso os novos valores lidos sejam diferentes dos anteriores \$scope é alterado e por meio de eventos que são disparados, a view é refrescada.

Para usar este *framework* é necessário fazer o *download* de uma cópia do ficheiro `angular.min.js` para a pasta `app/webroot/js`. Nesta diretoria foi criada outra pasta denominada `angularjs` onde foram colocados os ficheiros `app.js` e `angular-route.js` (também este baixado previamente) e criada uma pasta, `gráficosView`, que contém os restantes controladores. Dentro da view todos estes ficheiros são inicializados.

4.5 Cron Job

Um *cron job* permite agendar a execução de determinados ficheiros ou comandos. A sintaxe de um *cron* é “1 2 3 4 5 caminho/para/o/ficheiro” onde 1 representa os minutos, 2 as horas, 3 os dias, 4 os meses e 5 os dias da semana.

Os ficheiros que necessitam de serem executados automaticamente a intervalos ou datas pré-programadas são três: `meteo.php` e `tabelaQualidade_medias.php` na diretoria `app/webroot` e o ficheiro `ftp_LusitaniaGas.php` na diretoria `app/webroot/projeto`. O primeiro ficheiro é responsável por ler o *feed* de meteorologia e guardar o seu conteúdo na tabela *meteorologias* da BD “dashboard”. Apesar da alta frequência de atualização do *feed* (5 minutos) este ficheiro foi programado para ser executado a cada 15 minutos, por se considerar não haver interesse numa frequência de atualização mais elevada, com o seguinte código:

```
0,15,30,45, * * * * wget http://smarcity.deec.uc.pt/meteo.php
```

Já o ficheiro `tabelaQualidade_medias.php` é responsável por interpretar os valores de concentrações de gases poluentes e guardá-los na tabela *qualidade_medias* também da BD “dashboard”. O envio destes valores é feito sem qualquer frequência pré-estabelecida, e não à medida que são validados, embora exista um cálculo de médias por cada hora. Por esse motivo, optou-se por executar este ficheiro todas as horas:

```
0 * * * * wget http://smarcity.deec.uc.pt/tabelaQualidade_medias.php
```

Finalmente, o ficheiro `ftp_LusitaniaGas.php`, que guarda na tabela *gases* o volume de gás consumido no dia anterior é executado todos os dias pelas 8h, 9h, 10h e 11 horas. O motivo desta especificidade é a hora a que o ficheiro Excel que contém os valores de consumo de gás é atualizado pela Lusitaniagás. Embora oficialmente tenha sido comunicado que tal aconteceria todos os dias pela 7h30, verificou-se que por vezes esta atualização acontece depois das 10 horas, motivo pelo qual se alargou a janela de tempo em que o ficheiro é executado.

```
* 8 * * * wget http://smarcity.deec.uc.pt/projeto/ftp_LusitaniaGas.php
* 9 * * * wget http://smarcity.deec.uc.pt/projeto/ftp_LusitaniaGas.php
* 10 * * * wget http://smarcity.deec.uc.pt/projeto/ftp_LusitaniaGas.php
```

4.6 Gestão do desenvolvimento

Como foi descrito anteriormente, inicialmente o desenvolvimento do projeto foi feito apenas a nível local. Mais tarde, no entanto, foi feita uma migração do projeto para o servidor web.deec.uc.pt, para poder testar as funcionalidades do Dashboard em construção fora de um servidor local. Daí em diante, sempre que foram feitas alterações ao projeto a nível local foi necessário fazê-las chegar ao novo servidor. Para tal usou-se o sistema Git que, com um pequeno conjunto de comandos, “git add .”, “git commit”, “git push” e “git pull”, permite fazer chegar as alterações ao seu destino. Na imagem seguinte é descrito o processo segundo o qual funciona todo este sistema.

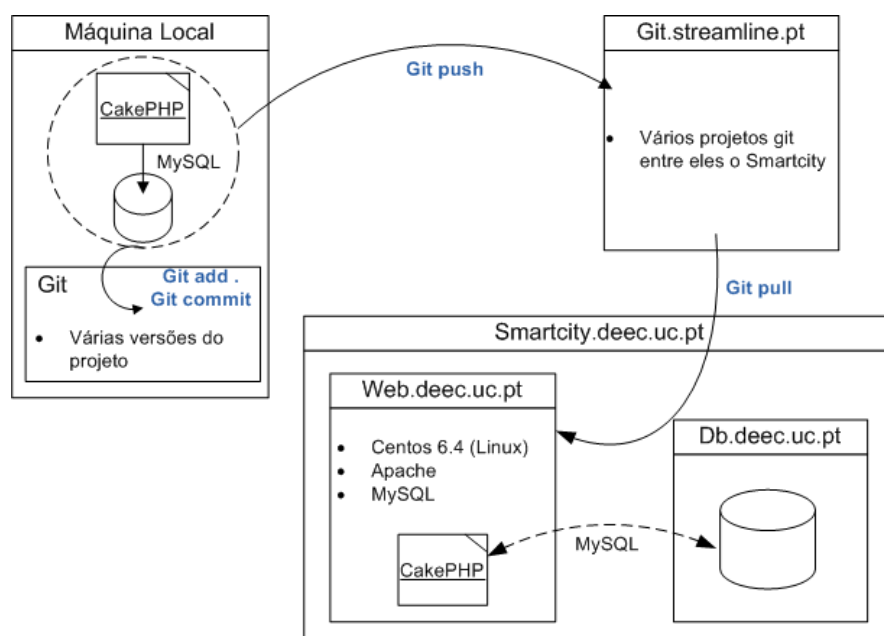


Figura 10 – Esquema da gestão e desenvolvimento do Dashboard

Sempre que surge uma alteração que deva ser guardada, com recurso ao programa Git Bash é executado o comando “git push”. Este comando carrega a nova versão do projeto para um servidor Git, o git.streamline.pt, onde existem outros projetos para além do Smarcity. Antes de ser executado este comando, são executados dois outros: o “git add .” e o “git commit”. O comando “git add” comunica ao Git que poderá haver novos ficheiros no projeto, em seguida utiliza-se o “git commit” para indicar verbalmente o que foi modificado.

```
git add .  
...  
git commit -m "nome da nova versão do projeto"  
...  
git push origin master
```

Figura 11 – Representação dos comandos usados em Git Bash

Com estes três comandos é criada uma nova versão do projeto com um nome designado pelo utilizador, e todas as versões ficam armazenadas por ordem cronológica no Git e disponíveis no servidor Git. Neste ponto as novas alterações podem ser visualizadas no servidor Git com o interface GitLab.

Para que a última versão fique disponível no servidor `Smartcity.deec.uc.pt` é necessário executar o comando “git pull” com o programa Putty. O Putty faz acesso remoto a este último servidor via SSH, permitindo executar comandos naquele servidor remotamente. É este método que faz a atualização do Dashboard Coimbra ao longo do processo de desenvolvimento.

Na primeira migração do projeto para o novo servidor foi usado o comando “git clone”. Ao executar este comando é feita uma cópia de todo o repositório do projeto. Deste ponto em diante o uso dos comandos descritos acima irá carregar apenas os ficheiros alterados ou adicionados.

5. Conteúdo do Dashboard

Uma vez detalhado o processo geral de desenvolvimento do SmartCoimbra Dashboard importa detalhar os procedimentos associados a cada um dos 14 painéis. A disposição dos painéis, como discutido anteriormente, decorre de variadas circunstâncias, entre as quais a importância do seu conteúdo e frequência de atualização mas também do modelo usado como inspiração para este Dashboard, o CityDashboard: London. Neste dashboard, o destaque é dado às informações com maior frequência de atualização, tais como a meteorologia, o estado de serviço dos meios de transporte, nível do rio, seguidas de painéis cuja informação é menos relevante por ser facilmente encontrada noutros locais, tais como as notícias, as tendências do Twitter e eventos.

Seguindo esta lógica, o SmartCity Coimbra Dashboard foi ganhando corpo. O conteúdo da página foi dividido em cinco partes, cada uma delas subdividida em 2 ou 3 colunas, consoante o tamanho requerido para cada painel, num total de 14 painéis. De seguida é descrito todo o processo de elaboração do conteúdo de cada um desses painéis, pela ordem em que estão dispostos no Dashboard (de cima para baixo, da esquerda para a direita).

5.1 Painel Meteorologia

Os valores de meteorologia, provenientes da estação meteorológica do DEM, são fornecidos por meio de um *feed* XML. Para ler estes valores foi criado um ficheiro PHP denominado `meteo.php` na diretoria `app/webroot` que acede à linha que contém os valores da seguinte forma:

```
$url = "http://www2.dem.uc.pt/meteo/rss/weewx_rss.xml";
$xml = simplexml_load_file($url);
$valores = $xml->channel->item->xpath('content:encoded');
$valor1=$valores[0];

$htmlParser = new DOMDocument();
$htmlParser->loadHTML($valor1);
$html = simplexml_import_dom($htmlParser);
$href = $html->body->p[0];
$linha=explode("\n", $href);

$horafinal= preg_split('/\s+/', explode("Time:", $linha[1])[1]);
$temperatura = (preg_replace("/^[^0-9]/", "", explode("Outside Temperature:",
$linha[2])[1]))/10;
...
```

Após a leitura, alguns destes valores estão prontos a serem guardados na base de dados, outros há que precisam de sofrer algumas alterações. Foram então criadas algumas funções neste

documento, tais como a função `direccao($graus)` que recebe como argumento a direção do vento em graus e devolve o seu sentido em pontos cardeais, colaterais ou subcolaterais.

O ícone que irá representar o estado do tempo atual deve estar de acordo com a hora do dia. Por exemplo, caso o estado do tempo seja “céu limpo” o ícone deverá mostrar um sol se estivermos entre a hora de nascer do sol e a hora de pôr-do-sol ou uma lua em caso contrário. Por esse motivo foi criado um código que, com base em três parâmetros, devolve o nome do ícone a usar tendo em conta a hora atual. Estes três parâmetros são o estado do tempo (`$imagem`), o momento do nascer do sol (`$nascer`) e o momento do pôr-do-sol (`$por`). Os três valores são obtidos de outro *feed*, pois a estação meteorológica do DEM não tem capacidade de interpretar as suas leituras e com base nelas prever o estado atual do tempo. Já a hora de nascer e pôr-do-sol são valores previamente calculados e portanto não são medidos nesta estação meteorológica. O *feed* onde são obtidos estes parâmetros pertence ao Openweathermap, e a sua leitura é feita de forma semelhante ao *feed* anterior.

O parâmetro `$estado`, onde é guardada a informação do estado do tempo, é usada numa segunda função denominada `traducao($estado)`, para traduzir o estado do tempo, que é fornecido em inglês, para português. O objetivo desta tradução é mostrar por escrito o estado do tempo atual ao passar o ponteiro do rato sobre o ícone do estado do tempo.

Após chamadas as duas funções e lidos todos os valores necessários ao painel Meteorologia é preciso armazená-los na tabela *meteorologias*. O processo pelo qual este armazenamento é feito, com recurso ao comando `INSERT INTO` na linguagem SQL, já foi descrito no capítulo anterior.

O painel Meteorologia é um dos painéis dinâmicos e como tal é usado o AngularJs para fazer a atualização automática dos valores. Foi criado o controlador 'meteoController' em `app.js` e nesse controlador é executado o ficheiro `meteo_bd.php`. com recurso a `$http.get`. O ficheiro `meteo.php` lê a última linha da tabela *meteorologias*, com recurso ao comando `SELECT` em SQL e envia os valores recuperados por uma tabela JSON sempre que é executado:

```
$meteo[] = array('temp' => $temperatura, 'humidade' => $humidade, 'vento' => $vento, 'direccao'=>$direccao,'pressao'=> $pressao, 'estado_tempo'=>$estado_tempo, 'imagem' => $imagem, 'chuva' => $chuva, 'radiacao' => $radiacao, 'uv' => $uv);
```

Dentro do controlador, em `app.js`, é usado o serviço `$scope` para recuperar e nomear os valores enviados da seguinte forma:

```
$http.get('/projeto/meteo_bd.php')
    .success(function(data) {
        $scope.meteo=data;
    });
```

Este código existe dentro da função `metedata()` que, com recurso ao serviço `$interval`, é executada todos os minutos:

```
var repeatFunction = $interval(metedata, 60000);
```


Para exibir os valores de meteorologia no painel é preciso chamar o módulo `ng-controller="meteoController"` dentro do *template*, no elemento HTML correspondente ao painel e posteriormente, no elemento HTML correspondente à tabela desse painel chamar o módulo `ng-repeat="x in meteo track by $index"`. Este módulo é necessário pois os valores de meteorologia estão guardados num *array* ou tabela chamada *meteo* (ver código anterior). Para mostrar o valores basta inserir o código `{{x.Nome_do_valor}}` em cada uma das células da tabela meteorologia.


Meteorologia					
Temp.	23.6 °C	Humidade	59 %	Índice UV	0
Vento	3.1 m/s	Direção	NNE	Radiação	136 W/m ²
Pressão	1015 hPa	Precip.	0 mm/h	Atual	

Figura 12 – Painel Meteorologia

Para além dos valores exibidos, é possível passar o ponteiro do rato sobre o ícone de Atual e obter a descrição do estado do tempo, como por exemplo “Céu Limpo”.

5.2 Painel Nível do Rio

Neste painel são exibidos os níveis hidrométricos do rio Mondego medidos nas estações Ponte Santa-Clara e Açude-Ponte. Até ao momento da conclusão desta dissertação não foi concretizado o envio dos valores de nível do rio Mondego por parte do SNIRH pelo que os valores exibidos neste painel são fictícios, lidos da tabela *Registres* da BD “mewago” .

Em *app.js* foi criado o controlador *mondegoController*, e nele, dentro da função *alturaMondego()* são usados os serviços `$http.get` e `$scope` para executar e ler os valores enviados pelo ficheiro *alturaMondego.php*. Este ficheiro encontra-se na diretoria *app/webroot/projeto*. A função *alturaMondego()* que engloba todo este processo é executada todos os minutos com recurso ao serviço `$interval`. Depois, dentro da tabela criada em HTML no *template* é inserido o módulo `ng-controller="mondegoController"` e os dois valores são chamados com `{{nome_estação}}`

Nível do Rio	
Estação	(m)
Açude-Ponte	1
Santa-Clara	0.8

Figura 13 – Painel Nível do Rio

5.3 Painel Qualidade do Ar

O terceiro painel do Dashboard é preenchido com informações relativas às concentrações de poluentes na cidade de Coimbra, que são fornecidas pela Divisão do Ambiente da CCDR pelo método GET.

O comando enviado contém no total 18 valores. São eles “acao” que se refere ao tipo de ação específica do envio, “id” e “password” usados para validar o envio, “id_estacao” que indica a que estação se refere o envio, “dia_ref_fim”, “mes_ref_fim”, “ano_ref_fim” e “hora_fim” que contém informação sobre a data e a hora da análise, “co8h”, “no2”, “o3”, “pm10” e “so2” que se referem às concentrações, “dia_ult_val”, “mes_ult_val”, “ano_ult_val” e “hora_ult_val” que contém a informação sobre a data e hora a que forma validados os valores pela última vez e finalmente “status”, que indica se os valores recebidos estão ou não validados.

Para receber os envios foi preparado o ficheiro `inserir.php` na diretoria do projeto `smartcoimbra/app/webroot` e é acedido pelo ATMIS pelo endereço `smartcity.deec.uc.pt/inserir.php`, que recebe a *string* de valores enviada e a armazena na base de dados. Uma vez que estamos a lidar com dois conjuntos de dados diferentes (um de cada estação), optou-se por construir duas tabelas, uma denominada *qtrafechos* que armazena os valores da estação de tráfego e outra denominada *qualidades* que armazena os valores da estação de fundo.

No início do script é aberta a ligação à base de dados, de forma semelhante às explicadas anteriormente e posteriormente são lidos os valores recebidos com o comando `$_GET`, com é mostrado no exemplo seguinte:

```
$id_estacao = $_GET['id_estacao'];
```

Para guardar os valores na respetiva tabela da base de dados existe uma condição *if* que separa os valores consoante o id da estação. Dentro de cada *if* é feito o armazenamento dos valores na sua tabela correspondente com o comando `INSERT INTO` em SQL.

Os valores enviados são uma réplica dos valores enviados para a APA. Caso a APA detete algum erro de leitura, envia uma resposta codificada que varia de 0 a 12, indicando à Divisão do Ambiente a natureza do erro. Em resposta, são enviados novamente todos os valores corrigidos, tanto para a APA como para o Dashboard. Por esse motivo, a retorno enviado pelo Dashboard é sempre 0 que é o equivalente a “valores corretos”.

Todos os valores de concentrações enviados correspondem a médias horárias, mas algumas das concentrações exibidas e usadas no cálculo do índice de qualidade do ar correspondem a médias octohorárias (caso do monóxido de carbono) e médias diárias (caso do PM₁₀). Por esse motivo, foi criada uma nova tabela na base de dados denominada *qualidade_medias* e um novo ficheiro denominado `tabelaQualidade_medias.php`. Com base nas concentrações médias horárias

lidas nas tabelas *qtrafechos* e *qualidades*, são calculadas as médias octohorárias e diárias dos respetivos poluentes (ver Apêndice B). As novas médias calculadas juntamente com as médias horárias dos restantes poluentes são gravadas na nova tabela com o seguinte código:

```
$sql6 = "INSERT INTO dashboard.qualidade_medias (co8h_trf, pm10_trf, pm10_fundo, no2_fundo, so2_fundo, o3_fundo, no2_trf, data_hora) VALUES ('$media_co8h', '$media_pm10', '$media_pm10_fundo', '$no2_fundo[$b]', '$so2[$b]', '$o3[$b]', '$no2[$a]', '$data_hora2')";
```

A frequência de envios de dados por parte Divisão do Ambiente não é exata e quase sempre são enviados valores repetidos. Por exemplo, às 7h da manhã de um determinado dia podem ser reenviados todos os valores existentes desse dia e até do dia anterior. Isto resulta em que por vezes cheguem a existir até 56 entradas nas tabelas *qualidades* e *qtrafechos* ao invés de apenas 24, uma por hora. Este facto, assim como o facto de as atualizações para as duas estações serem feitas em horários diferentes, dificulta o cálculo das médias dos valores, uma vez que é necessário pesquisar por todas as horas individualmente (uma pesquisa por id ou pelas linhas correspondentes a um dia não funcionaria por haver sempre dados repetidos), até chegar à última hora para a qual existem dados disponíveis em ambas as tabelas, *qualidades* e *qtrafechos*. Este foi outro dos motivos para criação de uma nova tabela, para facilitar a obtenção dos últimos valores disponíveis.

Também para o painel “Qualidade do Ar” foi criado um outro ficheiro, *Qualidade_Valores.php*, responsável por não só recuperar a última linha de valores da tabela *qualidade_medias* mas também por fazer o cálculo do Índice de Qualidade do Ar. Por se tratar de um processo longo e que recorre a muitas funções e que depende das concentrações de poluentes das duas estações, a sua explicação é detalhada no Apêndice B. É também em *Qualidade_Valores.php* que é feito o arredondamento para a unidade das concentrações recebidas, uma vez que estas vêm com 4 casas decimais, com a função `round()`.

Por se tratar de uma tabela com valores dinâmicos é usado o AngularJs no tratamento dos mesmos. Foi criado o controlador '*qualidadeController*' em *app.js* e nesse controlador é executado o ficheiro *Qualidade_Valores.php* com recurso a `$http.get`. Os valores enviados por JSON desse ficheiro são os seguintes:

```
$qualidade= array('no2_fundo' => $no2_fundo_1, 'o3_fundo' =>$o3_fundo_1, 'pm10_fundo'=>$pm10_fundo_1, 'so2_fundo' => $so2_fundo_1, 'co8h_trf' => $co8h_1, 'no2_trf'=>$no2_trf_1, 'pm10_trf'=>$pm10_trf_1, 'indice'=>$indicegeral);
```

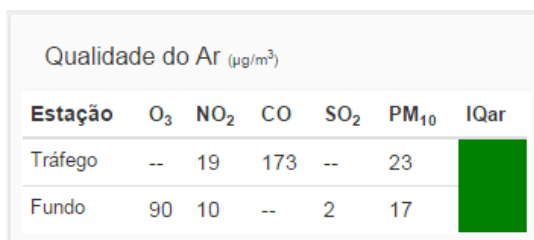
Dentro do controlador o serviço `$scope` é usado para recuperar o valor da seguinte forma:

```
$http.get('/projeto/Qualidade_Valores.php')
    .success(function(data, response) {
        $scope.no2Fundo = data.no2_fundo;
        $scope.o3Fundo = data.o3_fundo;
        ... })
```

Todo o código está envolvido pela função `Qualidade()` que, com o serviço `$interval`, é executada a cada meia-hora:

```
var repeatFunction = $interval(Qualidade, 1800000);
```

Dentro do *template*, após usar o módulo `ng-controller="qualidadeController"`, cada valor é mostrado no seu respetivo local da seguinte forma: `{{nome_do_poluente}}`.



Estação	O ₃	NO ₂	CO	SO ₂	PM ₁₀	IQar
Tráfego	--	19	173	--	23	
Fundo	90	10	--	2	17	

Figura 14 – Painel Qualidade do Ar

Algumas células da tabela estão vazias, preenchidas por “- -”. Isto acontece porque nem todos os poluentes são medidos nas duas estações. O monóxido de carbono, por exemplo, é um poluente que resulta diretamente da combustão de combustíveis fósseis e portanto a sua concentração é medida na estação de tráfego situada na Avenida Fernão de Magalhães, mas não na estação de fundo.

O resultado do índice é uma gama de 5 cores que representa o estado da qualidade do ar em toda a região de Coimbra. O número mínimo de valores necessários para o seu cálculo implica que este índice seja calculado pela primeira vez por volta das 15h de cada dia, pelo que até lá o que é visível no painel é a cor cinzenta e a indicação, por *tooltip*, que o índice ainda não foi calculado.

Outro uso do *tooltip* neste painel é sobre o nome dos poluentes, que indica o seu nome por extenso e o tipo de média a que se refere.

5.4 Painéis Energia Elétrica, Água e Gás

Estes três painéis têm aspetos muito semelhantes pois foram construídos sobre (quase) inteiramente o mesmo código. Por esse motivo este capítulo será subdividido em quatro partes, o primeiro onde se mostra como os diferentes valores são exibidos na forma de gráfico e os três últimos onde se explica como se obtém os valores para cada um dos painéis.

5.4.1 Highcharts

No início da construção deste novo protótipo do Dashboard foi usada a biblioteca Highcharts para construir os três gráficos principais. No entanto, à medida que o protótipo evoluiu e se tornou necessário fazer a atualização automática dos gráficos a intervalos de tempo predefinidos, tornou-se aparente que teríamos de usar o AngularJs e que o Highcharts só por si não respondia de forma esperada.

Existe um módulo em Angular-Js para o uso da biblioteca Highcharts chamado highcharts-ng. Com a inclusão deste novo módulo e com pequenos ajustes ao código inicial dos gráficos foi possível exibir gráficos dinâmicos. A inclusão do módulo highcharts-ng no angular é feita na definição do módulo principal, o chamado “modulo”, no início do ficheiro app.js:

```
var modulo =angular.module('dashboard', ['highcharts-ng']);
```

Cada um dos gráficos é codificado num controlador específico. Os três controladores são chamados de graficoEnergiaController, graficoAguaController e graficoGasController e existem em ficheiros JavaScript com o mesmo nome na diretoria app/webroot/js/angularjs/graficosView. Nessa mesma diretoria está presente o ficheiro highcharts-ng.js. Para exemplificar o código usado na construção destes controladores será usado o graficoEnergiaController:

```
angular.module('dashboard')
.controller('graficoEnergiaController', function ($scope,$http,$interval,$document) {
    var valores = function() {
        $http.get('/projeto/consumoEnergiaOntem.php')
            .success(function(data) {
                $scope.valor=data;
                $http.get('/projeto/consumoEnergiaHoje.php')
                    .success(function(data) {
                        $scope.valor_atual=data;
                    });
            });
    };
});
```

Após definir o novo controlador no início do ficheiro, cada controlador possui uma função denominada “valores”. Dentro dessa função, com recurso aos serviços \$http e \$scope são retirados os valores de dois ficheiros: consumoEnergiaOntem.php e consumoEnergiaHoje.php. Cada controlador lê dois ficheiros, embora com diferente nomenclatura, Agua em vez de Energia para o controlador graficoAguaController e Gas em vez de Energia para o controlador graficoGasController.

Se o sufixo do ficheiro for “Ontem”, então esse ficheiro vai buscar todo o consumo feito no dia anterior ao atual, por ordem cronológica, à correspondente tabela da base de dados. Da mesma forma, se o sufixo do ficheiro for “Hoje” o ficheiro recupera todos os valores de consumo do dia atual, até ao momento. A *array* com os valores do dia anterior chama-se \$scope.valor enquanto a que contem os valores de hoje é a \$scope.valor_atual.

A partir daqui o gráfico é definido depois de inicializado com o nome highchartsNG (\$scope.highchartsNG = { ... }). Entre as principais definições estão o tipo de gráfico, o tamanho, o tipo de *zoom*, a legenda, o *tooltip*, as cores e o tipo de dados apresentados. O código completo de um gráfico está presente no apêndice A. Para que haja uma atualização automática da representação gráfica dos valores, com recurso ao serviço \$interval, a função ‘valores’ é executada a cada x milissegundos., consoante a natureza dos valores. No caso dos valores de consumo de energia elétrica, a função é executada a cada 300 000 milissegundos ou 5 minutos:

```
var repeatFunction = $interval(valores, 300000);
```

Para exibir os gráficos é necessário em primeiro lugar inicializar dentro do *template*, onde é definido o código HTML para o painel correspondente, o controlador com recurso ao módulo ng-controller e depois chamar o respetivo gráfico:

```
<div class="panel-body" ng-controller="graficoEnergiaController">  
<highchart id="chart1" config="highchartsNG" ></highchart></div>
```

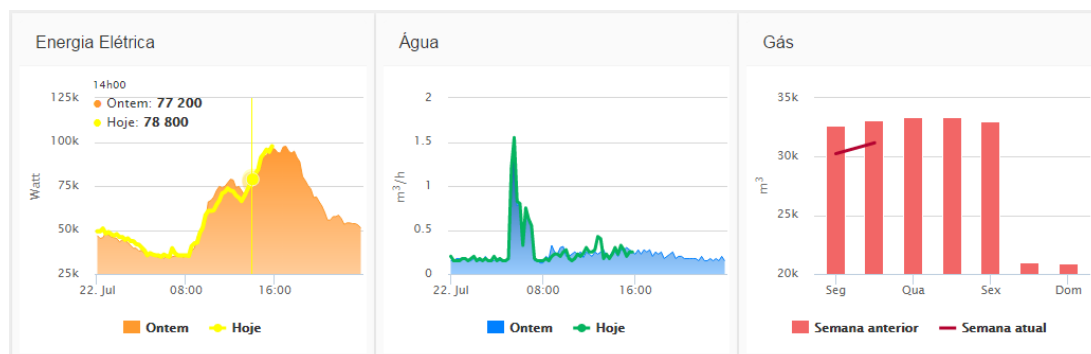


Figura 15 – Painéis Energia Elétrica, Água e Gás

Com as opções definidas para os gráficos é possível obter a representação gráfica do consumo do dia anterior, sobreposta pelo consumo do dia de hoje. É também possível obter informação mais detalhada ao passar com o ponteiro do rato sobre o gráfico, assim como clicar numa das legendas em baixo para esconder o respetivo gráfico. Os gráficos têm ainda a opção de *zoom* temporal, bastando para isso clicar numa qualquer zona do gráfico e arrastar e depois largar numa zona diferente. É possível fazer *zooms* sucessivos e para reverter basta clicar em “*zoom out*”, um botão que surge no canto superior direito sempre que se faz *zoom*.

Embora os valores representados nos gráficos Energia Elétrica e Água sejam provisórios, tudo nestes painéis está preparado para exibir os valores pretendidos quando estes forem finalmente facultados pelas respetivas empresas. Nessa altura basta alterar nos ficheiros `consumoEnergiaOntem.php`, `consumoEnergiaHoje.php`, `consumoAguaOntem.php` e `consumoAguaHoje.php` o nome das bases de dados e das tabelas para as que contém os novos valores.

5.4.2 Painel Energia Elétrica

Os valores exibidos no gráfico do painel Energia Elétrica são fornecidos pela empresa Streamline e correspondem ao consumo de energia elétrica do Departamento de Engenharia Eletrotécnica e de Computadores. Estes valores são conseguidos com o acesso à tabela *Registers* da base de dados “mewago”.

A tabela é atualizada a cada 15 minutos com novos valores de consumo de energia elétrica e para os adquirir foram criados dois ficheiros, o `consumoEnergiaOntem.php` e o

consumoEnergiaHoje.php. Ambos os ficheiros fazem ligação à base de dados e pesquisam pelo 'channel_id' 2, que corresponde à coluna que armazena os valores de consumo de energia elétrica. O primeiro ficheiro recupera todos os valores do dia anterior, num total de 56 valores, enquanto o segundo ficheiro procura apenas os valores correspondentes ao dia atual. No final, ambas as *arrays* de valores são enviadas por JSON para poderem ser lidas e interpretadas dentro do controlador graficoEnergiaController.

Como já foi mencionado na secção anterior, a frequência a que estes ficheiros são executados dentro da função 'valores' para o caso da Energia Elétrica é de 5 minutos.

5.4.3 Painel Água

À semelhança do painel anterior, também os valores de consumo de água são disponibilizados pela empresa Streamline e correspondem ao consumo do Departamento de Engenharia Eletrotécnica e de Computadores. Estes valores estão também presentes na tabela *Registers* da base de dados "mewago", mas por sua vez existem na coluna 'channel_id' com o número 37. A sua frequência de atualização é igualmente de 15 minutos.

Os ficheiros criados para a recuperação dos valores de consumo de água da base de dados chamam-se consumoAguaOntem.php e consumoAguaHoje.php. O primeiro procura por todos os valores correspondentes ao dia anterior, num total de 56, enquanto o outro ficheiro busca apenas os dados correspondentes ao dia atual. Em ambos os casos, no final do ficheiro os valores são enviados numa *array* JSON.

Dentro do controlador graficoAguaController, a função 'valores' é executada a cada 5 minutos após a primeira chamada da função.

5.4.3 Painel Gás

Os valores de consumo de gás da cidade de Coimbra são disponibilizados pela Lusitaniagás e são enviados para o Dashboard com o protocolo FTP. Neste envio é entregue uma folha Excel que contém, em diferentes separadores, as informações de consumo de gás de 6 zonas da região de Coimbra. A folha Excel corresponde ao mês em questão é atualizada automaticamente todos os dias entre 7h30 e as 10h30, altura em que é acrescentada uma nova linha em cada separador. Cada linha é composta por 4 colunas, a primeira com a data e a quarta, a que queremos, com o valor de gás consumido.

Para descarregar este ficheiro todos os dias, e para se fazer a sua leitura foi criado o ficheiro ftp_LusitaniaGas.php, a ligação FTP é feita com as seguintes linhas de código:

```
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to $ftp_server");  
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);
```

O armazenamento da folha Excel na mesma diretoria do ficheiro ftp_LusitaniaGas.php é feito com o seguinte código:

```
if (ftp_get($ftp_conn, $local_file, $server_file, FTP_ASCII)){
    echo "Successfully written to $local_file."; }
else{
    echo "Error downloading $server_file.";}

```

Após descarregar com sucesso a folha Excel é preciso ler a última linha dos separadores que contêm as zonas de Coimbra para que se possam armazenar na correspondente tabela da base de dados. Para este efeito, antes de mais é necessário instalar uma biblioteca capaz de interpretar os dados de uma folha Excel. A biblioteca escolhida é a PHPExcel e para sua instalação requer-se a cópia da pasta “Classes” para a diretoria app/webroot do CakePHP. Para interpretar os dados da folha Excel é preciso invocar a nova biblioteca.

```
require_once dirname(__FILE__) . '/../Classes/PHPExcel/IOFactory.php';

```

Em primeiro lugar é necessário procurar por um dos separadores necessários pelo seu nome, lê-lo, descobrir o número de linhas e colunas para posteriormente guardar o valor presente na última célula da última linha e coluna:

```
$sheetname = 'PRM CONDEIXA';
$objReader->setLoadSheetsOnly($sheetname);
$objPHPExcel = $objReader->load($filename);
$sheet = $objPHPExcel->getSheet(0);
$highestColumn = $objPHPExcel->setActiveSheetIndex(0)->getHighestColumn();
$highestRow = $objPHPExcel->setActiveSheetIndex(0)->getHighestRow();
for ($row = 1; $row <= $highestRow; $row++){
    $CondeixaRow = $sheet->rangeToArray('A' . $row . ':' . $highestColumn . $row,
        NULL, TRUE, FALSE);
}
$condeixa=$CondeixaRow[0][3];

```

Este procedimento é repetido para todos os separados que abrangem a zona de Coimbra, num total de 6 vezes. No final é feita a soma de todos esses valores, é aberta a ligação à base de dados e o valor é guardado com o comando INSERT INTO em SQL na tabela *gases*.

Após obtermos os valores necessários é imprescindível apagar o ficheiro para que no dia seguinte não ocorram problemas ao guardar uma nova folha Excel atualizada com os novos valores:

```
if (!unlink($file)){
    echo ("Error deleting $file");}
else{echo ("Deleted $file");}

```

Dentro do controlador *graficoGasController* com os serviços *\$http.get* e *\$scope* são executados e lidos os dois ficheiros criados para obter os valores da tabela *gases*. O ficheiro *consumoGasSemanaAnterior.php* lê os 7 valores anteriores à data da última segunda-feira, enquanto o ficheiro *consumoGasSemanaAtual.php* lê todos os valores desde a última segunda-feira. Ambas as tabelas de valores recuperadas são enviadas por JSON.

Dentro do controlador a função ‘valores’ é executada com uma frequência inferior às restantes, uma vez por hora, uma vez que os valores são atualizados apenas uma vez por dia.

5.5 Painel Consumo de Recursos

Neste painel é feita uma análise aos valores exibidos nos gráficos anteriores e são mostrados alguns indicadores. Por se tratar mais uma vez de valores dinâmicos, foi criado um novo controlador para cada um dos recursos representados, denominados `CEnergiaController`, `CAguaController` e `CGasController`. Cada um dos destes controladores executa uma função denominada `AteAoMomento()`, que, no seu interior, de forma semelhante à dos controladores anteriores, lê os valores de um ficheiro PHP com recurso aos serviços `$http.get` e `$scope`.

O ficheiro `Energia_Valores.php` lido pelo primeiro controlador acede à tabela `Registers` e lê todos os valores correspondentes ao consumo de energia elétrica do dia atual. Nessa `array`, é procurado o valor máximo e mínimo com as funções `max()` e `min()`, é feita a soma de todos os valores presentes no `array` e finalmente é calculado o seu tamanho original com `sizeof()`. Esses quatro valores são enviados por JSON no final do ficheiro. O ficheiro `Agua_Valores.php` é lido no segundo controlador, e, à semelhança do ficheiro anterior, lê da tabela `Registers` todos os valores de consumo de água do dia atual e com eles calcula a sua soma, máximo, mínimo e tamanho. Lê também da mesma tabela os consumos de água do dia anterior e calcula a sua soma. Com o valor de soma dos consumos do dia atual é ainda feita uma comparação com a capacidade de uma piscina média. Todos os seis valores são enviados por JSON. Finalmente, o último ficheiro `Gas_Valores.php` faz duas pesquisas na tabela `gases`, a primeira procura todos os consumos da semana atual e a segunda todos os consumos do período de tempo homólogo da semana anterior. Em cada um dos `arrays` é encontrado o valor máximo com a função `max()` e calculado o seu total. É também encontrado o valor de consumo mais recente desta semana e é feita a conversão do consumo do dia anterior para MW/h. No final, todos estes valores são enviados por JSON.

Uma vez lidos os valores dentro do controlador, alguns são ainda usados para calcular outros indicadores. No controlador `CEnergiaController` é calculada a potência média e o fator de carga enquanto que no controlador `CAguaController` a média de consumo do dia atual e anterior. Finalmente no controlador `CGasController` são calculadas a média de consumo de gás da semana atual e do período homólogo da semana anterior.

Para exibir estes valores é preciso chamar cada um dos controladores com o módulo `ng-controller` e depois, no seu local exato, usar `{{ nome_do_indicador }}`.



Figura 16 – Painel Consumo de Recursos

De forma a diferenciar as três diferentes naturezas das informações são usados três separadores, usados na mesma ordem dos painéis Energia Elétrica, Água e Gás. Dentro de cada separador existe um slide que a cada 4 segundos muda de conteúdo. Estas duas representações, os separadores e o slide, foram conseguidos com recurso a dois componentes JavaScript do Bootstrap, o ‘tab’ e o ‘carousel’ respetivamente. Para se fazer uso destes componentes foram copiados os ficheiros tab.js e carousel.js, provenientes da estrutura do Bootstrap, para a diretoria app/ webroot/ js, e invocados no topo do *template* do projeto, templateDashboard.ctp. Cada um dos slides recebeu uma cor diferente, consoante a natureza dos valores que estão a representar.

5.6 Painel Produção de Resíduos

Este painel é um dos poucos que usa valores estáticos no Dashboard, mas ainda assim é um dos mais dinâmicos em termos de interação para os utilizadores. Isto porque se optou por exibir os valores de recolha de resíduos de cada ano na forma de um gráfico de pizza ou diagrama circular com “*drill-down*”. O “*drill-down*” consiste em fazer uma exploração em diferentes níveis de detalhe das informações, o que neste caso permite dentro do mesmo gráfico obter outros gráficos, com diferentes tipos de resíduos.

Os três gráficos, um por cada ano, foram construídos com o Highcharts e estão divididos por separadores. Apesar de os valores serem estáticos optou-se por usar a ferramenta AngularJs também na construção deste painel e foi criado o controlador graficoResiduosController no ficheiro graficoResiduosController.js na diretoria app/ webroot/ js/ angularjs/ graficosView. No topo desse ficheiro são definidos em primeiro lugar os valores em dois *arrays* diferentes para cada ano: RESIDUOS_ANO e RESIDUOS_NIVEL2_ANO. O primeiro *array* contém os níveis superiores de tipo de resíduos, ou seja aqueles que vemos de imediato no Dashboard (“Indiferenciados” e “Outros Resíduos”) enquanto o segundo *array* contém todos os outros níveis inferiores que podem ser explorados (“Indiferenciados”, “Outros Resíduos”, “Ecopontos” e “Fora dos Ecopontos”). O critério de seleção dos níveis foi a quantidade em toneladas de cada

um, pois tipos de resíduos recolhidos em pequenas quantidades ficariam “invisíveis” se exibidos no mesmo gráfico com outros tipos de resíduos com quantidades recolhidas muito superiores.

O controlador é criado de forma semelhante aos anteriores, e é-lhe atribuído o mesmo nome do ficheiro. Dentro deste são definidos três variáveis diferentes com os nomes ‘chart2013’, ‘chart2012’ e ‘chart2011’:

```
angular.module('dashboard')
.controller('graficoResiduosController', ['$scope', function($scope) {
    var chart2013 = { ... }
    ...});
```

É dentro de cada uma dessas variáveis que são escolhidas as opções para cada um dos gráficos, sendo todos iguais à exceção dos valores contidos em cada um. Para exibir os gráficos é preciso usar o módulo ng-controller com o nome do controlador e depois, dentro de cada um dos separadores, usar o seguinte código com o nome da variável que define cada um dos gráficos:

```
<highchart config="chart2013" class="span10"></highchart>
```

O gráfico de pizza de cada ano está dividido em diferentes cores, cada uma delas com o respetivo nome. Ao passar com o ponteiro do rato sobre cada um é possível visualizar no canto inferior esquerdo do painel o nome do “nível” onde nos encontramos, o nome do tipo de resíduos que estamos a indicar e o número de toneladas correspondentes. Os tipos de resíduos que permitem aceder a um nível inferior estão sublinhados e ao abrir esse nível surge no canto superior direito um botão com a designação “Voltar” que permite regressar ao nível anterior.

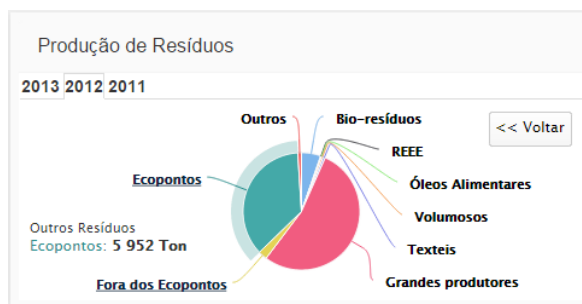


Figura 17 – Painel Produção de Resíduos

5.7 Painel Emissões de CO₂

Este painel mostra as emissões estimadas de dióxido de carbono resultantes da produção de energia elétrica, mais precisamente, da produção não renovável respeitante à queima de carvão e gás natural.

Para além da estimativa das emissões de dióxido de carbono global, o objetivo é mostrar também as emissões estimadas resultantes diretamente do abastecimento de água à cidade de Coimbra, razão pela qual existem duas colunas denominadas “Energia” e “Água”. Para esse efeito era necessário o índice de consumo, em kW/m³, para depois, com base no volume de água

consumido estimar o consumo de energia elétrica. De momento não temos o índice de consumo e portanto são usados valores fictícios no cálculo das emissões da coluna “Água”.

Para além das emissões estimadas resultantes do consumo do dia atual são também estimadas as emissões do dia anterior. Sendo a taxa de atualização dos valores de energia elétrica de 15 minutos, foi necessário recorrer ao AngularJs para fazer a atualização das emissões estimadas do dia atual. No ficheiro app.js foi criado o controlador EmissoesController que, com os serviços \$http.get e \$scope interpreta as emissões estimadas calculadas no ficheiro Emissoes.php, que se encontra na diretoria app/ webroot/ projeto. Este ficheiro vai, em SQL, recolher os valores de consumo de energia elétrica do dia atual e anterior da tabela *Registers* da base de dados *mewago*. Após se efetuarem os cálculos, os 8 valores são enviados por JSON. Dentro do controlador, a função ‘emissoes’ que engloba todo este código é executada todos os minutos com o auxílio do serviço \$interval.

Emissões de CO ₂		
Tipo	Energia	Água
Carvão hoje(kg)	198.177	3.965
Gás hoje(kg)	3703.2	0.908
Carvão ontem(kg)	299.577	5.994
Gás ontem(kg)	68.576	1.372

Figura 18 – Painel Emissões de CO₂

Para exibir as estimativas foi criada uma tabela e os valores são mostrados da mesma forma que os restantes valores em AngularJs até aqui: chamar o módulo ng-controller com o nome do controlador e escrever o nome das variáveis calculadas dentro de duas chavetas no local apropriado.

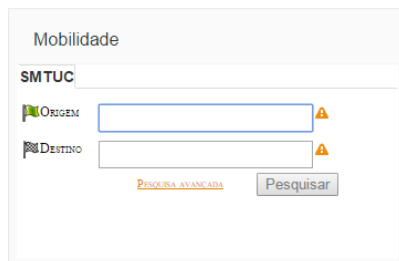
5.8 Painel Mobilidade

De futuro, a ideia é que figurem neste painel vários simuladores de percurso de diferentes categorias de transportes utilizados dentro da cidade de Coimbra. Por esse motivo, o painel foi concebido para albergar diferentes conteúdos no mesmo espaço divididos por separadores. No entanto, visto que até ao momento da conclusão deste projeto apenas a empresa SMTUC aceitou colaborar com este projeto, existe apenas um separador denominado ‘SMTUC’.

A construção deste separador é semelhante aos mencionados anteriormente e o seu conteúdo é ocupado por um *iframe* disponibilizado, na fase anterior de desenvolvimento, pela empresa SMTUC:

```
<iframe src="http://rumosweb.smtuc.pt/rumos/default.aspx" frameborder="0" scrolling="no" height="143" style="width:100%"></iframe>
```

O *iframe* permite fazer uma pesquisa de itinerários e horários, escolhendo diretamente a ‘Origem’ e o ‘Destino’ ou simplesmente clicando em ‘Pesquisa avançada’. Em qualquer uma das situações o utilizador é redirecionado, dentro do separador do *browser*, para o novo destino, o *website* da SMTUC. Como foi mencionado na secção anterior, esta situação não é ideal e espera-se que possa ser melhorada no futuro.



Mobilidade

SMTUC

ORIGEM

DESTINO

[PESQUISA AVANÇADA](#)

Figura 19 – Painel Mobilidade

5.9 Painel Eventos

O conteúdo deste painel é preenchido por um *iframe* disponibilizado pelo *website* da Agenda7. Esta agenda tem uma tipologia mensal, na qual estão marcados os eventos a decorrer em cada dia do mês:

```
<iframe src='http://agenda7-coimbra.pt/agenda7/embedCalendar_v2' width='200' height='180' frameBorder='0'></iframe>
```

Este *iframe*, apesar de ter sido modificado especialmente para o Dashboard, ficou muito aquém do pretendido. Ainda assim, é uma agenda funcional para pesquisa dos eventos, embora a sua maior limitação seja estar programada para, ao clicar num desses eventos, o *website* da Agenda7 abrir no próprio painel do Dashboard.

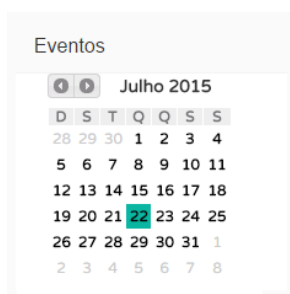


Figura 20 – Painel Eventos

5.10 Painel Locais de Interesse

Neste painel é exibido um mapa da cidade de Coimbra construído utilizando a plataforma *online* “Google Maps Engine”. Com esta ferramenta foi possível construir um mapa personalizado que assinala os diferentes pontos de interesse listados na secção anterior. O mapa é depois incluído no painel através de um *iframe*:

```
<iframe src='https://mapsengine.google.com/map/embed?mid=ztZj1NV0pbFY.kMiimrs9dA_k&z=13' height='250' style='width:100%'></iframe>
```

Os pontos estão divididos em 6 cores diferentes, consoante a sua natureza: Verde para jardins e espaços verdes; Bordô para Museus e Monumentos; Amarelo para as subestações de Energia Elétrica; Azul claro para as estações de medição de qualidade do ar; Castanho para a estação de meteorologia; Azul para a estação de tratamento de água, ETA, Boavista II.

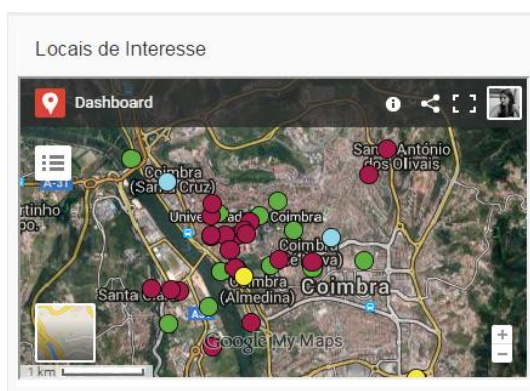


Figura 21 – Painel Locais de Interesse

Ao clicar sobre qualquer um dos pontos assinalados o nome desse local surge sobre o mesmo. É também possível clicar no ícone quadrado que se assemelha a uma lista, do lado esquerdo, que abre a legenda das diferentes cores. O mapa permite também fazer “zoom in” ou “zoom out” no canto inferior direito ou clicar sobre qualquer ponto do mapa e arrastá-lo para que se possam ver outras zonas.

5.11 Painel Notícias

As notícias exibidas neste painel são obtidas de dois *websites* de jornais, o Diário de Coimbra e o Destakes que alimentam respetivamente dois *feed* RSS de notícias. Recorreu-se à ferramenta gratuita *online* “Feed Wind” que converte um ou vários URL’s de *feed*’s num *widget* que é depois embutido no projeto. Ao escolher mais do que um *feed* estamos a garantir que, mesmo que um deles falhe, haja sempre outro *feed* a fornecer notícias:

```
(function () {
    var params = {
        rssmikle_url: "http://local.destakes.com/coimbra/?format=rss|http://www.noticiasdecoimbra.pt/feed/",
        rssmikle_frame_width: "100%",
        rssmikle_frame_height: "250",
        rssmikle_target: "_blank",
        rssmikle_font: "Arial, Helvetica, sans-serif",
        ...
    };
    feedwind_show_widget_iframe(params);
})();
```

O código para este *widget* está presente dentro de um ficheiro JavaScript na diretoria `app/webroot/js` com o nome `feednoticias.js`. Este ficheiro é invocado dentro do painel da seguinte forma:

```
<script src="/js/feednoticias.js"></script>
```

Esta ferramenta permite também personalizar a aparência do *widget*. A informação é mostrada com o título, texto parcial da notícia, data e hora da mesma. Ao clicar sobre uma notícia o utilizador é redirecionado para a notícia completa no *website* responsável por ela.



Figura 22 – Painel Notícias

As notícias surgem dentro de uma caixa que faz “*auto-scroll*” a cada 3 segundos. No entanto, ao posicionar o ponteiro do rato sobre ela surge uma barra de “scroll” do lado direito e é possível navegar por todas as notícias. A frequência de atualização do *widget* é de 15 minutos.

5.12 Painel Farmácias

O painel Farmácias é preenchido por um *iframe* disponibilizado gratuitamente pelo *website* Farmácias de Serviço.net que informa o utilizador sobre quais as farmácias disponíveis nesse momento no concelho de Coimbra:

```
<iframe src="http://farmaciasdeservico.net/widget/?localidade=coimbra%7Ccoimbra&... " height="250" style="width:100%"  
frameborder="0" ></iframe>
```

Para além do nome da farmácia, é também indicada a morada e contacto telefónico e também o tipo de disponibilidade: “Permanente” para o caso de farmácias abertas permanentemente; “Horário alargado” seguido do respetivo horário de abertura no presente dia; “Disponibilidade” para as farmácias abertas no momento, em horário normal. A informação disponibilizada é atualizada todos os dias às 9 horas.



Figura 23 – Painel Farmácias

A caixa é acompanhada de um “scroll” do lado direito que permite a navegação por todas as farmácias assim como de uma hiperligação no final designada “Mais informações” que ao ser clicada redireciona o utilizador para o *website* Farmácias de Serviço.

5.13 Modal

A acompanhar cada um dos painéis existe um texto informativo respeitante ao tipo de informação exibida por ele e à forma como foi conseguida. Este texto, por vezes acompanhado por tabelas, surge quando o utilizador clica sobre o título do painel, num espécie de *pop-up* que se sobrepõe à restante página. Isto é conseguido com o componente JavaScript ‘modal’ do Bootstrap. Para o usar foi necessário copiar o ficheiro modal.js para a diretoria app/ webroot/ js e invocá-lo no início do templateDashboard.ctp. Na pasta Vendor, existente na pasta app, foi criado o ficheiro modal.php e é dentro dele que estão todos os textos e tabelas, criados em HTML, exibidas nos *pop-ups*. Este ficheiro contém duas funções, a primeira denominada ‘conteúdo’ onde estão contidas as informações de cada painel dentro de condições *switch* e a função ‘modal’. A função ‘modal’ recebe com argumento o id e nome do modal, chama a função ‘conteúdo’ e devolve todo o código HTML referente à criação do modal e o seu conteúdo. A função ‘modal’ é chamada dentro do *template*, no espaço de código referente ao título de cada um dos painéis:

```
<div class="panel-heading">  
  <div class="panel-title" style="cursor: pointer" data-toggle="modal" data-target="#myModal"> Meteorologia</div>  
  <?php modal('myModal','Meteorologia') ?>  
</div>
```

Ao clicar sobre o título de qualquer um dos painéis, surge um segundo painel a deslizar do topo enquanto o restante painel é escurecido. Para voltar ao Dashboard basta clicar na cruz no topo direito do ‘modal’ ou clicar em qualquer zona fora do mesmo. No Apêndice C é possível consultar os textos informativos de todos os painéis.

5.14 Versão Final

Apresentados todos os painéis que figuram no Dashboard é importante mostrar o aspeto final do mesmo. Como foi relatado em secções anteriores, pretendia-se obter um Dashboard funcional de aparência sóbria e principalmente informativo.

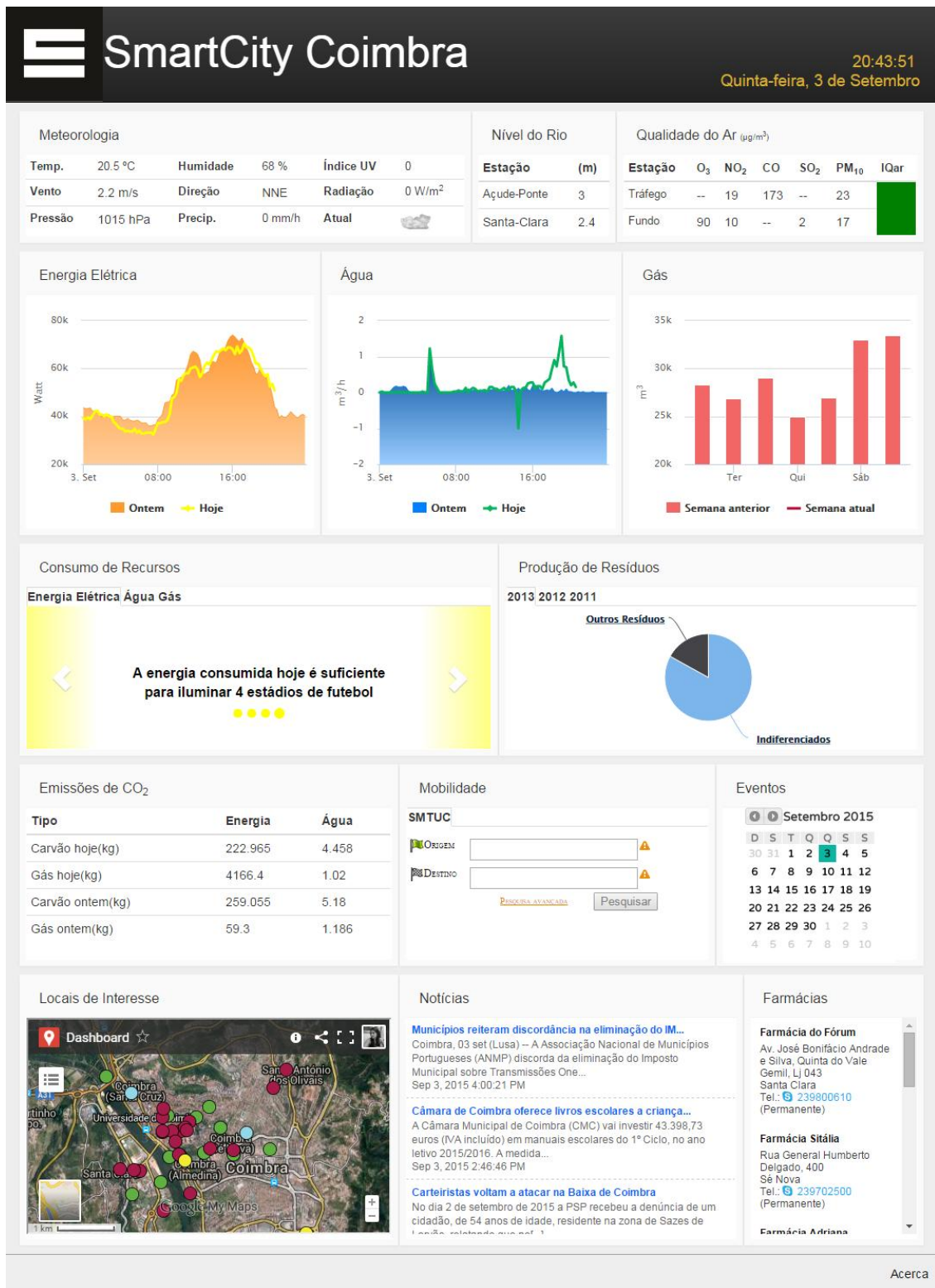


Figura 24 – Aspetto final do Dashboard

Ao longo desta dissertação foram relatados os diferentes métodos usados para a recolha das informações necessárias a cada painel assim como as dificuldades associadas a cada um deles. Ainda assim, não foi possível concretizar todas as ligações às várias fontes de dados requeridas. Na figura seguinte está ilustrado o estado em que, no final desta versão do projeto, cada uma das recolhas de dados quantitativos, concretizadas ou não, se encontra.

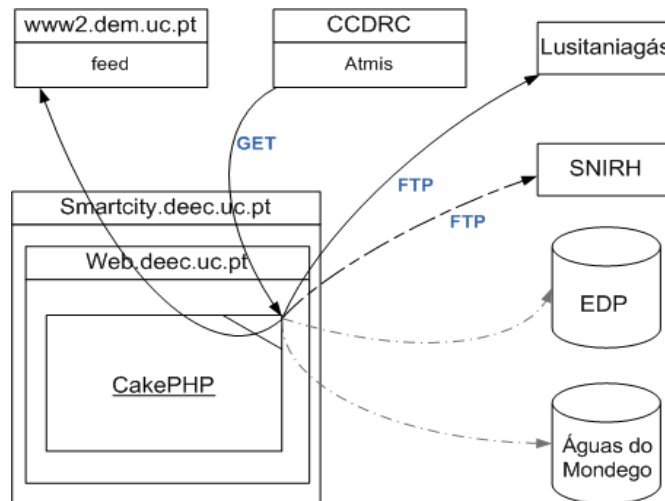


Figura 25 – Ligações entre o Dashboard e as várias fontes de informação

5.15 Versão Mobile

Um dos objetivos traçados no início do desenvolvimento da nova versão do SmartCoimbra Dashboard foi a construção de uma aplicação para *smartphones*. Com esse intuito, e fazendo uso das potencialidades do CakePHP foi criada uma versão *mobile* da plataforma Dashboard. Para começar, foi criado o ficheiro `mobile.ctp` na mesma diretoria do *template* do projeto, `app/View/Layouts`. Neste ficheiro foi inserido todo o código de `templateDashboard.ctp` e posteriormente, painel a painel, foram feitas as alterações necessárias para criar a aplicação. Para que o CakePHP fizesse a distinção entre qual dos dois ficheiros carregar consoante o dispositivo em que o Dashboard está a ser acedido, foi inserido no ficheiro `PagesController.php` o seguinte código:

```
public function display() {
    if ($this->request->is('mobile')) {
        $this->layout = 'mobile';
    } else {
        $this->layout = 'templateDashboard';
    }
    ...
}
```

Uma vez que foi necessário alterar a disposição, tamanho ou forma de alguns dos componentes do Dashboard, foi criada uma cópia do ficheiro `bootstrap.min.css` na diretoria `app/webroot/css` e renomeada para `bootstrap.min-mobile.css`. O novo ficheiro é chamado no início do novo *template*.

Começando pelo título, este sofreu as maiores alterações: a disposição do nome foi alterada, o tamanho de letra foi reduzida e a hora e data foram retiradas por se tornarem redundantes numa aplicação móvel. De seguida, foi a vez do esquema do conteúdo, pois na versão *mobile* só há espaço para um painel por linha. Após retirar todos os códigos que criavam as colunas foram

analisados todos os painéis para identificar quais necessitam de sofrer ajustes para se adaptarem à nova versão.



Meteorologia			
Temp.	23.3 °C	Humidade	49 %
Vento	4.5 m/s	Direção	NNE
Pressão	1018 hPa	Precip.	0 mm/h
Índice UV	0	Radiação	528 W/m ²
Estado Atual	☀️		

Nível do Rio	
Estação	Metros
Açude-Ponte	0.5
Santa-Clara	0.4

Figura 26 - Versão mobile do SmartCity Coimbra Dashboard

O painel Meteorologias foi alterado para conter duas colunas em vez de três e passou a ter cinco linhas. Na última linha foi acrescentado o estado do tempo à frente do ícone. No painel Rio, agora com mais espaço, foi possível alterar “(m)” para “Metros” e finalmente, o último painel a sofrer alterações foi o painel Consumo de Recursos, onde a largura mais reduzida obrigou a fazer alguns ajustes em termos de posição relativa de alguns dos seus componentes.

Também os componentes ‘modal’ sofreram alterações. No final de cada *pop-up* de informação foi acrescentado um botão “Fechar” para facilitar o encerramento dos mesmos. Para este efeito foi necessário criar o ficheiro modal – *mobile.php* na pasta Vendor e inseri-lo no cabeçalho.

6. Conclusão e sugestões de trabalhos futuros

Ao longo desta dissertação foi relatado o processo de elaboração de uma nova versão para o Dashboard do projeto SmartCoimbra, mais completa, funcional e apelativa do que o primeiro protótipo. Nesta nova versão é possível consultar informações de diversas categorias relativas à cidade de Coimbra, entre as quais a mobilidade, ambiente, consumo de recursos mas também a cultura, notícias e turismo. Durante o desenvolvimento foram ponderadas outras informações que seriam acrescentadas ao Dashboard mas que por diversos motivos não houve capacidade de incorporar até ao momento. Alguns exemplos dessas informações são as estimativas de emissões de CO₂ relativas ao tráfego de Coimbra, que poderão vir a ser conseguidas com um simulador desenvolvido do Departamento de Engenharia Civil, ou um bloco dedicado ao desporto em Coimbra uma vez que esta é uma cidade com uma grande população jovem e com muitos praticantes de desporto.

No decorrer de todo este processo de criação de um Dashboard para Coimbra alarguei os meus conhecimentos em desenvolvimento *web*, adquiri prática no domínio de diversas ferramentas e linguagens de programação mas também desenvolvi as minhas capacidades de comunicação num ambiente profissional, resultado dos contactos com as diversas entidades mencionadas nesta dissertação.

A plataforma desenvolvida pode ser consultada *online* no endereço smartcity.deec.uc.pt. No entanto, este é apenas um endereço provisório uma vez que o objetivo final é o de o Dashboard ser associado ao projeto SmartCoimbra e ser lançado num dos servidores da CMC.

Sendo este um projeto ambicioso e estando as TIC em constante transformação é expectável que o Dashboard, e a própria SmartCoimbra, evoluam futuramente. O progresso do Dashboard poderá consistir na inclusão de novas informações, adição de novas funcionalidades, renovação de *design*, entre outras. Como sugestões de desenvolvimentos futuros identificam-se os seguintes pontos:

- Aquisição automática dos valores referentes à energia elétrica, água consumida e resíduos recolhidos;
- Adição de um painel que exiba as estimativas de emissões de CO₂ relativas à circulação automóvel na cidade de Coimbra;
- Adição de um painel sobre os eventos desportivos em Coimbra;

- Adição de um painel que exiba e promova aplicações móveis submetidas por cidadãos de Coimbra, mediante aprovação prévia;
- Conseguir uma atualização em horários pré-estabelecidos dos controladores criados com o AngularJs;
- Contacto com as entidades responsáveis pela projeto SmartCoimbra para lançamento da nova versão nessa plataforma;
- Elaboração de uma versão em língua inglesa, para visibilidade internacional do projeto.

Referências

- [1] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. A. Pardo, H. J. Scholl, "Understanding Smart Cities: An Integrative Framework", in 45th Hawaii International Conference on System Sciences, 2012.
- [2] S. AlAwadhi, H. J. Scholl, "Aspirations and Realizations: The Smart City of Seattle", in 46th Hawaii International Conference on System Sciences, 2013.
- [3] S. Suakanto, S. H. Supangkat, Suhardi, Saragih, "Smart City Dashboard for Integrating Various Data of Sensor Networks".
- [4] Arup's IT and Communications Systems, "Smart Cities Transforming the 21st century city via the creative use of technology", September 2010.
- [5] Oberlin Environmental Dashboard [Online]. Available : <http://environmentaldashboard.org/>.
- [6] London Dashboard [Online]. Available : <http://citydashboard.org/london/>.
- [7] City Dashboard Amsterdam [Online]. Available : <http://amsterdamsmartcity.com/>.
- [8] Green Addict Bristol [Online]. Available : <http://www.greenaddict.eu/>.
- [9] Follow Aarhus [Online]. Not Available : <http://citydashboard.alexandra.dk/Dashboard/>.
- [10] Green Energy Dashboard [Online]. Available : <http://greenenergy.reno.gov/energy/>.
- [11] Agência Internacional de Energia [Online]. Available : <http://www.iea.org/statistics/resources/unitconverter/>
- [12] Twitter Bootstrap [Online]. Available : <http://getbootstrap.com/>.
- [13] David Golding, "Beginning CakePHP From Novice to Professional", 2008 [Online]. Available : https://books.google.pt/books?id=9_tn7ogNFo4C&lpg=PA138&ots=X2yB3k_Dyf&dq=cakephp%20function%20that%20is%20called%20every%20few%20minutes&hl=pt-PT&pg=PP1#v=onepage&q=cakephp%20function%20that%20is%20called%20every%20few%20minutes&f=true.
- [14] "Build a forum with CakePHP (part 1)" [Online]. Available : <http://www.startutorial.com/articles/view/build-a-forum-with-cakephp-part-1>.
- [15] AngularJs Tutorial [Online]. Available : https://docs.angularjs.org/tutorial/step_00.

Apêndice A – Manual técnico de Manutenção

Neste apêndice serão descritos as principais características e componentes usados das principais ferramentas utilizadas na criação do SmartCity Coimbra Dashboard. É também detalhada a estrutura base do esquema deste painel informativo e as suas propriedades.

CakePHP

Este *framework* de desenvolvimento rápido para PHP, tem como principal objetivo permitir um desenvolvimento de uma aplicação web de forma rápida e estruturada, sem perder a flexibilidade. É livre e de código aberto e utiliza o padrão de arquitetura MVC (Model, View e Controller).

Para fazer uso do CakePHP é preciso fazer o *download* de uma cópia do *website* oficial para a diretoria do servidor local que se pretende usar, neste caso Xampp/ htdocs/ projects pois é o Xampp o servidor escolhido para o desenvolvimento na máquina local. Depois basta começar a adicionar e modificar o conteúdo do interior desta aplicação.

A estrutura de arquivos do CakePHP tem o seguinte aspeto:

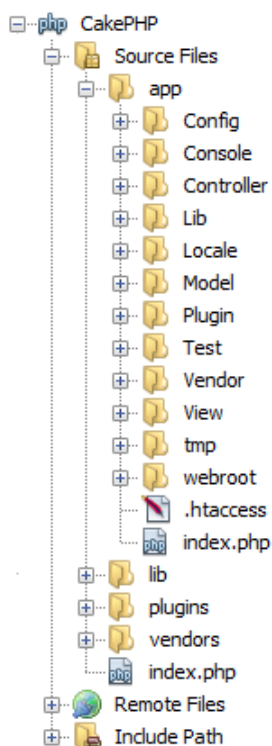


Figura 27 – Estrutura de arquivos do CakePHP

As principais diretorias têm os seguintes propósitos:

- Config: Configuração da app (base de dados, configuração do núcleo, etc);
- Controller: Contém os controladores da app. Fornecem uma série de métodos, sendo estes chamados de ações. Os Controladores são usados para gerir a lógica da aplicação;
- Locale: Strings de internacionalização;
- Model: Contém os modelos da app. São responsáveis por representar dados, ou seja, são utilizados para fornecer acesso aos dados;
- Plugins: Contém os pacotes de plugins;
- Tmp: local de arquivos temporários;
- Vendor: aplicativos de terceiros;
- View: Contém as Views (Vistas) do sistema, o *template* e o Layout. São responsáveis por gerar a interface gráfica ao utilizador. Estes arquivos (Views) são guardados com a extensão “.ctp” e escritos em php;
- Webroot: raiz da aplicação. Contém todos os ficheiros extras adicionados à aplicação, imagens, etc.

É na diretoria “View”, dentro da pasta “Layout” que está presente o *template* da aplicação. Nesta diretoria, já se encontra definido o layout padrão (default.ctp) que foi renomeado para templateDashboard.ctp.

Um *template* deve conter uma região destinada para carregar os arquivos da view. Isto é executado através da função `fetch('content')`. Para incorporar arquivos externos ao *template*, deve-se utilizar a função `fetch('script')` e definido o nome dos ficheiros que serão importados.

Dentro da diretoria “Webroot” foi criada uma nova pasta denominada “Projeto”. Dentro desta pasta estão todos os ficheiros criados durante o desenvolvimento desta aplicação. Dentro da diretoria “Webroot” foi também copiado o arquivo “Classes”, uma biblioteca PHPEXcel que permite trabalhar folhas de cálculo do Microsoft Excel com a linguagem PHP.

A arquitetura MVC (Model, View e Controller), sob a qual o CakePHP opera, é uma abordagem de *software* que separa a lógica da aplicação da sua apresentação. Na prática, ela permite que as aplicações *web* contenham o mínimo de script já que a apresentação é separada do código. O seu funcionamento é descrito de seguida.

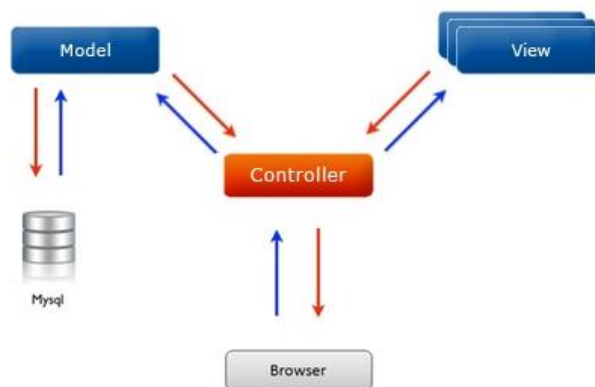


Figura 28 – Funcionamento da arquitetura MVC

- O *browser* envia requisição à aplicação;
- O controller processa a ação;
- O controller processa a requisição e acessa a lógica de negócio (model) da aplicação;
- O model acessa a base de dados da aplicação para executar a ação requisitada pelo controle;
- O controller então seleciona a view correspondente a ação;
- O controller renderiza a view incluindo os dados de resposta da ação;
- O controller devolve a resposta para o *browser*;

Base de Dados

O MySQL é um servidor e gestor de base de dados SQL, tendo como principais características, a rapidez de manipulação de dados, multitarefa, multiusuário, e robustez. É usado na maioria das aplicações *web* baseadas em PHP. O MySQL é de código aberto e gratuito, é também multiplataforma, estando disponível para os principais sistemas operativos.

O MySQL é usado para fazer a comunicação entre o aplicativo *web* e a base de dados, ou seja, grava os dados introduzidos pelo usuário na BD, da mesma forma, caso o usuário queira fazer uma consulta é o MySQL que pesquisa as informações na base de dados. Portanto, a sua função é armazenar dados e disponibilizá-los quando são solicitados.

Neste projeto, foi usado o software “MySQL Workbench” da empresa Oracle, que auxiliou na conceção e organização das bases de dados. Foi criada a base de dados “dashboard” e nela foram inseridas 6 tabelas com as seguintes nomenclaturas: *meteorologias*, *qualidades*, *qtrafechos*, *rios*, *gases* e *qualidades_medias*.

Bootstrap

Bootstrap é uma coleção de vários elementos e funções personalizáveis para projetos da *web*, empacotados previamente numa única ferramenta. Esses elementos personalizáveis contidos no Bootstrap são uma combinação de HTML, CSS e JavaScript. Além disso, o Bootstrap possui uma diversidade de componentes (*plugins*) em JavaScript (jQuery) que permitem dinamizar os componentes da página *web*: tooltip, menu-dropdown, modal, carousel, slideshow, entre outros sem a menor dificuldade, apenas acrescentando algumas configurações no código, sem a necessidade de escrever mais linhas de código. No *website* da aplicação, <http://getbootstrap.com/>, é possível ver e copiar diretamente o código dos elementos necessários à aplicação em construção.

A instalação desta ferramenta é simples e consiste no *download* da cópia do pacote para a máquina local. A partir daqui é possível criar um novo *template* e personalizá-lo alterando o código CSS dos ficheiros bootstrap.min.css e bootstrap-theme.min.css e o código JavaScript do ficheiro bootstrap.min.js.

Ligação do CakePHP à Base de Dados

A ligação entre a aplicação *web* e a base de dados é feita no script denominado “database.php” que se encontra na diretoria Source Files/ app/ Config do CakePHP. Na figura seguinte encontra-se o código que faz essa ligação à base de dados (com nomes fictícios), dando a essa ligação o nome de “default”.

```
public $default = array(  
    'datasource' => 'Database/Mysql',  
    'persistent' => false,  
    'host' => 'my.host.pt',  
    'login' => 'my_login',  
    'password' => 'my_password',  
    'database' => 'database_name',  
    'prefix' => '',  
);
```

Posteriormente modificam-se os campos ‘login’, ‘password’ e ‘database’, respetivamente para o usuário de acesso à BD, palavra-passe e nome da BD.

Integração do Bootstrap no CakePHP

A integração da ferramenta Bootstrap na estrutura do CakePHP consiste na cópia de alguns ficheiros desse pacote para pastas específicas do CakePHP:

- Copiar os ficheiros bootstrap.min.css e bootstrap-theme.min para a pasta app/ webroot / css;
- Copiar o ficheiro bootstrap.min.js para a pasta app/ webroot/ js;

- Copiar a pasta “font” para app/ webroot.

Esta é a integração genérica do Bootstrap no CakePHP. Para incluir alguns dos componentes do Bootstrap usados é preciso copiar também as bibliotecas desses *plugins* JavaScript para a pasta app/ webroot/ js: modal.js, tab.js e caroussel.js.

Para fazer uso do Bootstrap basta agora inserir todos os códigos dos componentes pretendidos no *template* do projeto, na diretoria app/ view /layouts, e no seu cabeçalho incluir todos os ficheiros Bootstrap copiados:

```
<head>
  <?php
    echo $this->fetch('css');
    echo $this->fetch('script');
    echo $this->Html->script('modal.js');
    echo $this->Html->script('carousel.js');
    echo $this->Html->script('tab.js');
    echo $this->Html->css('bootstrap.min');
    echo $this->Html->css('bootstrap-theme.min.css');
  ?>
  ...
</head>
```

O ficheiro bootstrap.min.css é o que contém a maioria das definições dos elementos do Bootstrap, na linguagem CSS, e portanto é este o ficheiro no qual devemos fazer alterações para personalizar esses elementos. Para facilitar esta tarefa o código deste ficheiro foi separado em várias linhas, permitindo assim uma pesquisa mais rápida e eficaz dos excertos de códigos que necessitem de alterações.

Highcharts

O Highcharts é uma biblioteca de gráficos escritos inteiramente em JavaScript, que oferece uma forma fácil de adicionar gráficos interativos a uma página *web*. Na sua biblioteca estão disponíveis os mais variados tipos de gráficos, tais como área, linha, colunas, barras, entre outros. As grandes vantagens do Highcharts são: ser suportado em todos os navegadores modernos, não necessitar de plugins do lado do cliente, tais como Flash ou Java, e ter uma sintaxe de configuração simples.

Para usar esta biblioteca é necessário inicializá-la, neste caso no final do *template*, com o seguinte código:

```
<script src="http://code.highcharts.com/highcharts.js"></script>
<script src="http://code.highcharts.com/modules/exporting.js"></script>
<script src="http://code.highcharts.com/modules/drilldown.js"></script>

<script>
  jQuery(document).on( 'shown.bs.tab', 'a[data-toggle="tab"]', function (e) {
    jQuery( ".span10" ).each(function() {
      var chart = jQuery(this).highcharts();
      chart.reflow()
    });
  });
</script>
```

Uma vez que a segunda parte recorre ao jQuery, foi feito o *download* do ficheiro jquery.min.js para a diretoria app/ webroot/ js e inicializado no início do *template* com o código `$this->Html->script('jquery.min.js');`. O JQuery é uma biblioteca JavaScript para manipulação do DOM (ex. mudança de cores de um elemento, eventos onclick, animação).

Os gráficos são chamados com a tag `<Highchart>` e com o respetivo id:

```
<highchart id="chart1" config="highchartsNG"></highchart>
```

AngularJs

O AngularJs é um *framework* estrutural, desenvolvido em 2009 pela equipa da Google, que tem como objetivo simplificar o desenvolvimento de aplicações *web*. É escrito inteiramente em JavaScript e permite estender a sintaxe HTML de forma a exprimir os componentes das aplicações de forma clara e sucinta. Este *framework* é usado para permitir uma atualização de dados de forma automática e autónoma, sem a imposição de fazer o recarregamento de toda a página para visualizar os valores atualizados.

As principais características desta *framework* são o Data-binding bidirecional, o padrão de desenvolvimento MVC, a personalização de diretivas, injeção de dependências, comunicação com o servidor entre outras. A utilização deste *framework* permite simplificar quer a camada de desenvolvimento, quer obter resultados mais interessantes em termos de *User Experience* sem aumentar exageradamente a complexidade da solução ou os conhecimentos técnicos requeridos.

A aplicação AngularJs contém vários conceitos que separam diferentes partes da aplicação em construção e a organizam devidamente. Os principais conceitos usados nestes projeto são:

- MVC

Este padrão de desenvolvimento *web* é, tal como no caso do CakePHP, composto por três partes:

- Model ou Modelo: é o nível mais baixo do padrão e é responsável por gerir os dados;
- View: É responsável por exibir toda ou parte dos dados ao utilizador. A view corresponde ao *template* da aplicação, neste caso o ficheiro `templateDashboard.ctp`;
- Controller ou Controlador: contém o código que controla as iterações entre o model e a view. Os controladores do AngularJs são normalmente mantidos em ficheiros à parte.

- Diretivas

As diretivas ou “directives” permitem estender a linguagem HTML em respostas às necessidades da aplicação. O AngularJs vem com várias diretivas definidas, algumas das quais

foram usadas no projeto, tais como “ng-app”, “ng-controller”, “ng-repeat”, “ng-cloak”. As principais no entanto são a “ng-app” e “ng-controller”.

A diretiva “ng-app” é usada para inicializar a aplicação AngularJs e encontra-se no topo da view do projeto.

```
<html ng-app="dashboard">
```

A diretiva “ng-controller” define o controlador. Os controladores permitem, mediante programação, implementar a lógica de apresentação do AngularJs. Servem principalmente para dividir certas partes do código e neste projeto foi criado um controlador para cada painel que faça uso do AngularJs, num total de 11 controladores. Os controladores são criados em ficheiros à parte, sendo o principal o app.js, que são depois inicializados no ficheiro principal.

```
<div class="Meteo" ng-controller="meteoController">
```

- Data-Binding

O AngularJs tem a capacidade de fazer a ligação direta e bidirecional dos dados (Data Binding bidirecional) que permite a sincronização automática entre models e views. Desta forma é reduzida a quantidade de códigos para mostrar os dados processados pelo servidor.

- Módulos

No AngularJs, um módulo é uma coleção de componentes. Devem ser utilizados para agrupar objetos que contribuem entre si para algum objetivo e também para facilitar a reutilização de comportamentos. Assim, uma aplicação é composta por um ou mais módulos mas neste projeto existe apenas um módulo denominado “modulo” que é definido no ficheiro app.js.

```
var modulo =angular.module('dashboard', ['highcharts-ng']);
```

Na definição deste módulo podemos ver dois parâmetros: “dashboard”, o nome dado a esta aplicação com a diretiva “ng-app” e “highcharts-ng”. Este último é uma diretiva criada especificamente pelo Highchart, a biblioteca de gráficos usados neste projeto, para que se possam usar os seus gráficos em conformidade com o AngularJs.

- Controladores

Existe um controlador por cada painel. São eles denominados “modegoController” para o painel Nível do Rio, “meteoController” para o painel Meteorologia e finalmente um controlador para cada gráfico: “graficoEnergiaController”, “graficoAguaController” e “graficoGasController”. Os dois primeiros controladores estão definidos em app.js enquanto os

três últimos são definidos cada um num ficheiro com o mesmo nome. Todos estes ficheiros estão na diretoria `app/ webroot/ js`. Um controlador é definido do seguinte modo:

```
modulo.controller('meteoController', function($scope,$http,$interval) {...});
```

- Serviço

Um serviço é outro componente do AngularJs que é usado para organizar e partilhar código transversalmente à aplicação. São funções que responsáveis por executarem tarefas específicas e nesta plataforma foram usados 3 serviços: `$scope`, `$http` e `$interval`.

- `$scope`

O `$scope` é um serviço usado para conectar os controladores às views para as quais foram ligados com a diretiva “ng-controller”. Um controlador pode adicionar valores ou funções ao seu `$scope` que depois estarão visíveis na view.

```
modulo.controller('meteoController', function($scope,$http,$interval) {  
    ...  
    $scope.meteo=data;  
    ...  
});
```

No exemplo anterior, na view a variável “meteo” terá o valor que lhe é associado pelo `$scope` do controlador `meteoController`. Quando ocorrem alterações no `$scope`, é disparado um evento. Com estes eventos a view sabe quando deve refrescar a variável “meteo”.

- `$http`

O `$http` é um serviço do AngularJs usado para ler valores de um servidor remoto. Este serviço é fundamental neste projeto pois é com ele que são executados os ficheiros que adquirem os valores pretendidos da base de dados.

```
modulo.controller('meteoController', function($scope,$http,$interval) {  
    var meteodata = function() {  
        $http.get('/projeto/meteo_bd.php')  
            .success(function(data) {  
                $scope.meteo=data;  
            }).error(function(data) {  
                ...  
            });  
    };  
    ...  
});
```

Este serviço usa apenas um parâmetro, o url ou diretório do ficheiro que queremos aceder, que é usado para gerar um pedido http, e devolve uma “promise” (ou promessa), com dois métodos específicos: “success” (no caso de sucesso no acesso ao ficheiro) e “error” (no caso da ocorrência de algum erro).

- `$interval`

Este serviço do AngularJs é usado para controlar o intervalo de ação de uma determinada função. Os seus parâmetros são a função que está a controlar e o tempo, em milissegundos, entre cada iteração.


```
var repeatFunction = $interval(meteodata, 60000);
```

É com estes três serviços que é feita a atualização automática dos valores apresentados no Dashboard. O serviço `$http.get` lê os valores pretendidos a cada intervalo definido pelo serviço `$interval`. Caso os novos valores lidos sejam diferentes dos anteriores `$scope` é alterado e por meio de eventos que são disparados, a *view* é refrescada.

Para usar este *framework* é necessário fazer o *download* de uma cópia do ficheiro `angular.min.js` para a pasta `app/ webroot/ js`. Nesta diretoria foi criada outra pasta denominada `angularjs` onde foram colocados os ficheiros `app.js` e `angular-route.js` (também este baixado previamente) e criada uma pasta que contém os restantes controladores:

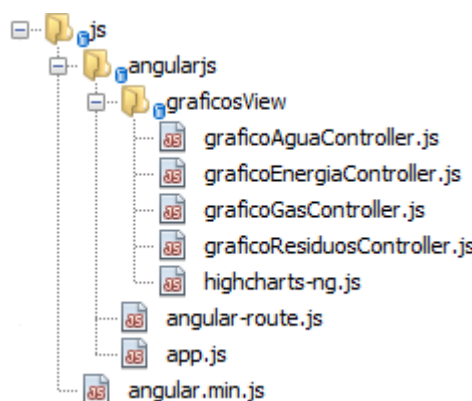


Figura 29 – Estrutura de arquivos do AngularJs

Alguns destes ficheiros são inicializados no top do *template* enquanto outros, para não atrasar o carregamento da página, são apenas inicializados no final deste:

```
<head>
  <?php
    echo $this->Html->script('angular.min.js');
    echo $this->Html->script('angularjs/graficosView/highcharts-ng.js');
    echo $this->Html->script('angularjs/angular-route.js');
  ?>
</head>

<!--Corpo do template-->

<?php
  echo $this->Html->script('angularjs/app.js');
  echo $this->Html->script('angularjs/graficosView/graficoGasController.js');
  echo $this->Html->script('angularjs/graficosView/graficoAguaController.js');
  echo $this->Html->script('angularjs/graficosView/graficoEnergiaController.js');
  echo $this->Html->script('angularjs/graficosView/graficoResiduosController.js');
?>
```

Layout

O *layout* do Dashboard é do tipo cabeçalho-conteúdo-rodapé. Estas divisões são feitas com recurso a componentes do Bootstrap, que são identificados com a *tag* `<div>`. Este elemento é utilizado frequentemente para o posicionamento absoluto do conteúdo.

```

<body role="document">
  <!-- Cabeçalho-->
  <div class="navbar navbar-inverse navbar-static-top" role="navigation">
  </div>
  <!-- Conteúdo da Página-->
  <div class="jumbotron">
  </div>
  <!-- Rodapé-->
  <div id="footer" >
  </div>
</body>

```

Dentro do cabeçalho existem outros componentes, um deles para o título e logotipo, outro para a data e hora. A data e hora são definidos num script chamado clock.js que está na pasta “js” da diretoria “webroot”, e é incluído no *template* com o código `$this->Html->script('clock')`. Dentro desse ficheiro é usado o método JavaScript `getElementById()` para definir um novo id denominado “clockbox”. Desta forma, dentro do *template* podemos chamar a hora e data pelo seu id:

```

<div id="clockbox" style="font:14pt Arial; color:#D4AF37; text-align: right"></div>

```

O conteúdo da página está subdividido em 14 painéis distribuídos por cinco linhas cada uma com duas a três colunas. As linhas são elementos do Bootstrap chamadas de “row” enquanto que as colunas são elementos cujo nome varia de “col-md-1” a “col-md-12” consoante o tamanho necessário a cada painel.

```

<div class="row">
  <div class="col-md-10"><!--//////////////////// Painel Meteorologia////////////////////////////////////-->
  <div class="panel panel-default">
    <div class="panel-heading">
      <div class="panel-title" style="cursor: pointer" data-toggle="modal" data-
target="#myModal">Meteorologia</div>
      <?php modal('myModal','Meteorologia') ?></div>
      <div class="Meteo" ng-controller="meteoController">
        <!-- ////Conteúdo do painel//// -->
      </div>
    </div>
  </div>
</div>

```

Dentro de cada painel, no elemento dedicado ao cabeçalho é chamada a função `modal()` com dois argumentos que assegura que ao se clicar no cabeçalho desse painel se abre uma janela (“modal”) com uma pequena redação sobre o conteúdo desse painel. O código da função `modal()` assim como as respetivas informações de cada painel encontram-se na diretoria Vendor sob o nome `modal.php` e este ficheiro é incluído no *template* com o código `App::import('Vendor', 'modal')`.

Finalmente o rodapé detém um *link* com o nome “Acerca”, que abre um “modal” que contém informações sobre este projeto, com descrição do projeto, fornecedores de dados, suporte técnico, contactos, etc.

```

<div id="footer" >
  <div class="container" style="background-color:#eee; margin: 1px 0 0; margin-color: transparent; height: 45px">

```

```

<a href="#myModal" data-toggle="modal" data-target="#myModalfinal" style="line-height:45px; vertical-align:
middle;cursor: pointer">Acerca</a>
<?php modal('myModalfinal','Acerca do SmartCity Coimbra Dashboard') ?></div></div>

```

A maioria das definições de estilo dos elementos usados está definida no ficheiro *bootstrap.min.css*. No entanto, como se pode verificar no excerto de código exibido em cima, por vezes essas opções são definidas dentro da própria `<div>` com recurso ao código `style=" "`.

Tabelas

Existem um total de 4 tabelas no Dashboard, todas elas conseguidas com o elemento ‘Table’ do Bootstrap. A composição básica de uma tabela é a seguinte:

```

<table class="table">
  <thead> <!--Cabeçalho da tabela -->
    <tr><!--Linha -->
      <th>Estação</th><!--Coluna do cabeçalho-->
      <th>Metros</th>
    </tr>
  </thead>
  <tbody> <!--Conteúdo da tabela -->
    <tr><!--Linha -->
      <td style="font-size: 10pt;">Açude-Ponte</td><!--Coluna -->
      <td ng-cloak>{{alturaAcude}}</td><!--Coluna -->
    </tr>
    <tr><!--Linha -->
      <td style="border-bottom: 2px solid white">Santa-Clara</td><!--Coluna -->
      <td ng-cloak>{{alturaSantaClara}}</td><!--Coluna -->
    </tr>
  </tbody>
</table>

```

Gráficos

Os gráficos são conseguidos com a biblioteca Highcharts, na linguagem JavaScript, e existem 4 no Dashboard: 3 são do tipo linha e área e quarto é do tipo ‘drill-down’. De seguida é mostrado o código completo de um dos gráficos de linha e área:

```

$scope.highchartsNG = {
  options: {
    chart: {
      type: 'line', //Tipo de gráfico
      events: {
        redraw: function() {
        }},
      spacingBottom: 8, //Espaçamento dos lados
      spacingTop: 30,
      spacingLeft: 0,
      spacingRight: 2,
      zoomType: 'x' //Tipo de Zoom
    },
    credits: {enabled: false}, //Permitir texto ou não
    exporting: {
      buttons: {
        contextButtons: {
          enabled: false,
          menuItems: null}},

```

```

        enabled: false},
legend: {enabled: true}, //Permitir legenda ou não

tooltip: { //informação que surge ao passar o ponteiro do rato
    backgroundColor: null,
    borderWidth: 0,
    shadow: false,
    useHTML: true,
    hideDelay: 0,
    shared: true,
    padding: 0,
    crosshairs: [ {
        width: 1,
        color: '#FFFF00',
        zIndex: 22
    } ],
    xDateFormat: '%Hh%M',
    headerFormat: '<span style="font-size: //cabeçalho do tooltip
10px">{point.key}</span><br/>',
    positioner: function () { //posição do tooltip
        return { x: 60, y: 0 };
    }
},
},
xAxis: {type: 'datetime', //formato do eixo x
    minRange: 900000, //em milissegundos: 15 minutos
    tickmarkPlacement: 'on',
},
yAxis: {title: {text: 'Watt'}}, //título do eixo y

series: [{ //Primeira curva do gráfico :Ontem
    color: '#FE9A2E',
    type: 'area', //tipo de curva
    name: 'Ontem', //nome da curva

    fillColor: { //cor do gráfico
        linearGradient: { x1: 0, y1: 0, x2: 0, y2: 1},
        stops: [
            [0, 'rgb(255, 153, 51)'],
            [1, 'rgb(255, 204, 153)']
        ]
    },
    marker: {
        radius: 2
    },
    lineWidth: 1, //espessura da linha
    states: {
        hover: {
            lineWidth: 1 //espessura da linha ao passar o rato
        }
    },
    threshold: null,

    pointInterval: 900 * 1000, //intervalo dos dados
    pointStart: Date.UTC(ano, mes, dia, hora, minuto, segundo),
    data: $scope.valor //valores que entram na curva: array com valores lidos
da BD
},
    { //Segunda curva do gráfico:Hoje
    color: '#FFFF00',
    type: 'line', //tipo de curva
    name: 'Hoje', //nome da curva
    lineWidth: 3, //espessura da linha
    pointInterval: 900 * 1000, //intervalo de dados
    pointStart: Date.UTC(ano, mes, dia, hora, minuto, segundo),
    data: $scope.valor_atual //valores que entram na curva: array com valores
lidos da BD
    }],
title: {
    text: '' //título do gráfico
},
},

```

```

loading: false,
size: {
    height: 270//altura do gráfico
}
}

```

Separadores

Os separadores são mais um elemento Bootstrap denominado por ‘nav-tabs’ e são usados 3 vezes no Dashboard. De seguida é mostrado o código de um ‘tab’ com 3 separadores:

```

<!-- Nav tabs -->
<ul class="nav nav-tabs" role="tablist"><!--Inicialização da lista de separadores -->
    <li role="presentation" class="active"><a href="#2013" aria-controls="home" role="tab" data-toggle="tab"
style="font-weight: bold; color: #333; border-left-color:transparent">2013</a></li><!--2013 é o nome deste separadaor -->
    <li role="presentation" ><a href="#2012" aria-controls="home" role="tab" data-toggle="tab" style="font-weight:
bold; color: #333">2012</a></li>
    <li role="presentation" ><a href="#2011" aria-controls="home" role="tab" data-toggle="tab" style="font-weight:
bold; color: #333">2011</a></li>
</ul>
<!-- Tab panes -->
<div class="tab-content"><!--Conteúdo dos separadores-->
    <div role="tabpanel" class="tab-pane fade in active" id="2013" ><highchart config="chart2013"
class="span10"></highchart></div>
    <div role="tabpanel" class="tab-pane fade" id="2012" ><highchart config="chart2012" class="span10"></div>
    <div role="tabpanel" class="tab-pane fade" id="2011" ><highchart config="chart2011" class="span10"></div>
</div>

```

Cada um dos separadores deve conter uma referência diferente. Essa referência, ‘href’ é usada para conectar o separador ao respetivo conteúdo. Cada contudo tem um ‘id’ que deve ser igual a um dos ‘href’ dos separadores. O elemento ‘active’ é usado para indicar qual o separador que está, por omissão, aberto.

Carousel

Um ‘Carousel’ é um elemento do Bootstrap usado para criar *slide-shows*. Este componente foi um dos últimos a ser incorporado no Dashboard e apenas porque era necessário mostrar várias informações dentro do mesmo painel sem que para isso se compromettesse a projeção das mesmas. Existem ao todo 3 slide-shows, um por cada separador do painel Consumo de Recursos. O código de um desses slides é o seguinte:

```

<div id="myCarousel" class="carousel slide" data-ride="carousel">
    <!-- Indicadores -->
    <ol class="carousel-indicators">
        <li data-target="#myCarousel" data-slide-to="0" class="active" style="border: 2px solid #FFFF00; background-
color: #FFFF00"></li>
        <li data-target="#myCarousel" data-slide-to="1" style="border: 2px solid #FFFF00; background-color:
#FFFF00"></li>
        <li data-target="#myCarousel" data-slide-to="2" style="border: 2px solid #FFFF00; background-color:
#FFFF00"></li>
    </ol>
    <div class="carousel-inner" role="listbox" ng-controller="CEnergiaController">
        <div class="item active"><!--Conteúdo da primeira página do slide-->
            <!--Imagem de fundo da
página -->
                <div class="container">
                    <div class="carousel-caption" >

```

```

        <p1 style="font-weight: bold;">Pontas do dia de Hoje<br></p1>
        <p1 ng-cloak>Valor Máximo: {{maximo}} W<br></p1>
        <p1 ng-cloak>Valor Mínimo: {{minimo}} W</p1>
    </div>
</div>
</div>
<div class="item"><!--Conteúdo da segunda página do slide-->
<img src='<?php echo $this->webroot, 'img/fundo1.png'; ?>' border=0' alt="imagem2" />
    <div class="container">
        <div class="carousel-caption">
            <p1 style="font-weight: bold" ng-cloak>Potência média<br> {{potencia}} W</p1>
        </div>
    </div>
</div>
<div class="item"><!--Conteúdo da terceira página do slide-->
<img src='<?php echo $this->webroot, 'img/fundo1.png'; ?>' border=0' alt="imagem3" />
    <div class="container">
        <div class="carousel-caption">
            <p1 style="font-weight: bold" ng-cloak>Fator de Carga<br> {{carga}}</p1>
        </div>
    </div>
</div>
<!--Controladores: botões para mudar de página -->
<a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev" style="background-image:
linear-gradient(270deg, rgba(255,255,255,1) 0%, rgba(255,255,0,1) 100%)">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#myCarousel" role="button" data-slide="next" style="background-image:
linear-gradient(to right, rgba(255,255,255,1) 0%, rgba(255,255,0,1) 100%)">
    <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>

```

O código das páginas dos slides devem ser colocadas no *template* pela ordem pretendida e a página que por omissão é exibida em primeiro lugar é identificada com `class="active"`. Cada slide tem um o seu id específico. Este slide tem o id de 'MyCarousel' enquanto os outros dois são identificados por 'MyCarousel2' e 'MyCarousel3'.

Apêndice B – Cálculo do Índice de Qualidade do Ar

O índice de qualidade do ar de Coimbra resulta da média aritmética calculada para cada um dos poluentes medidos nas duas estações, de tráfego e fundo. O tipo de média varia entre cada poluente:

- Dióxido de azoto (NO₂) - médias horárias
- Dióxido de enxofre (SO₂) - médias horárias
- Ozono (O₃) - médias horárias
- Monóxido de carbono (CO) - médias de 8 horas consecutivas
- Partículas inaláveis (PM₁₀) - média diária

Os valores enviados pela sistema Atmis vêm na forma de média horária pelo que o primeiro passo no cálculo do índice é calcular a média diária das PM₁₀ e a média octohorária do CO. No ficheiro `tabelaQualidade_medias.php` é estabelecida a ligação à base de dados 'dashboard' e com o comando SELECT, por cada novo envio correspondente a uma nova hora, são recuperados os últimos 8 valores de CO da tabela *qtrafechos*.

```
$$sql3 = "select * from dashboard.qtrafechos where data_hora <= '$data_hora2' order by id desc limit 8";
```

A tabela *qualidades* armazena os valores da estação de fundo e portanto não lê os valores de monóxido de carbono, pois este poluente está relacionado com as emissões de tráfego. Com os 8 valores recuperados é calculada a média octohorária:

```
$soma=0;
$a=0;
for($i = 0; $i < sizeof($co8h);$i++){
    if($co8h[$i] != 0){
        $a=$a+1;
        $soma=$co8h[$i]+$soma;
    }
}
$media_co8h=$soma/$a;
```

No mesmo ficheiro são recuperados de forma semelhante, por cada novo envio, todos os valores de PM₁₀ do dia atual das tabelas *qtrafechos* e *qualidades*. Com esses valores são calculadas duas médias diárias:

```
$soma2=0;
$a2=0;
$a=(sizeof($pm10)-$diferenca);
for($i = (sizeof($pm10)-1); $i >= $a; $i--){
    if($pm10[$i] != 0){
        $a2=$a2+1;
        $soma2=$pm10[$i]+$soma2;
    }
}
```

```

    }
}
$media_pm10=$soma2/$a2;
$soma3=0;
$a3=0;
$b=sizeof($pm10_fundo)-$diferenca;
for($i = (sizeof($pm10_fundo)-1); $i >= $b; $i--){
    if($pm10_fundo[$i] != 0){
        $a3=$a3+1;
        $soma3=$pm10_fundo[$i]+$soma3;
    }
}
$media_pm10_fundo=$soma3/$a3;

```

A variável `$diferenca` é importante pois por vezes são enviados valores repetidos e portanto é fundamental garantir que as médias não são afetadas. Os envios de valores repetidos, quando acontecem, contêm sempre os valores pela ordem correta e desde a meia-noite desse dia (os valores mais recentes são sempre os últimos da tabela). Por exemplo, se às 13 horas de um determinado dia existirem 22 entradas numa das tabelas precisamos apenas de recuperar os últimos 13 valores. Por esse motivo a variável `$diferenca`, que corresponde à diferença de horas entre a hora do último envio e a meia-noite do dia atual, é calculada da seguinte forma:

```
$diferenca=round((strtotime($data_hora2) - strtotime($meia_noite))/3600, 1);
```

Após serem calculadas todas as médias necessárias, estas e as restantes médias horárias dos outros poluentes são armazenadas, a cada novo envio, num nova tabela denominada *qualidade_medias* da seguinte forma:

```

$sql6 = "INSERT INTO dashboard.qualidade_medias (co8h_trf, pm10_trf, pm10_fundo,
no2_fundo, so2_fundo, o3_fundo, no2_trf, data_hora)
VALUES ('$media_co8h', '$media_pm10', '$media_pm10_fundo', '$no2_fundo[$b]',
'$so2[$b]', '$o3[$b]', '$no2[$a]', '$data_hora2')";

```

O segundo passo no cálculo do índice de qualidade do ar é determinar o número de médias disponíveis para cada poluente, pois para o cálculo deve existir um número mínimo de 11 medições por cada poluente. Se um poluente é medido nas duas estações então o seu número medições é a soma das medições nessas duas estações.

No ficheiro `Qualidade_Valores.php` é estabelecida a ligação à base de dados “dashboard” e são recuperados todos os valores do dia atual desde a meia-noite com recurso ao comando `SELECT` em SQL da tabela *qualidade_medias*. Depois é calculado um a um o número de medições totais de cada poluente:

```

$no2_indice[]=0;
$no2_media=0;
$o3_indice[]=0;
$o3_media=0;
$indice_no2=0;
for($i = 0; $i < sizeof($no2_fundo);$i++){
    if($no2_fundo[$i] != 0 && $no2_trf[$i] != 0){
        $indice_no2=$indice_no2+2;
        $no2_indice[$i]=$no2_fundo[$i]+$no2_trf[$i];
        $no2_media=$no2_indice[$i]+$no2_media;
    }
}

```



```

elseif($no2_fundo[$i] != 0 && $no2_trf[$i] = 0){
    $indice_no2=$indice_no2+1;
    $no2_indice[$i]=$no2_fundo[$i];
    $no2_media=$no2_indice[$i]+$no2_media;
}
elseif($no2_fundo[$i] = 0 && $no2_trf[$i] != 0){
    $indice_no2=$indice_no2+1;
    $no2_indice[$i]=$no2_trf[$i];
    $no2_media=$no2_indice[$i]+$no2_media;
}
}
}
for($i = 0; $i < sizeof($o3_fundo);$i++){
    if($o3_fundo[$i] != 0){
        $o3_indice[$i]=$o3_fundo[$i];
        $o3_media=$o3_indice[$i]+$o3_media;
    }
}

```

No excerto anterior é exemplificado como se determina o número de médias de um poluente que é medido nas duas estações (NO₂) e de um que é medido apenas numa estação (O₃). Para se determinar se um poluente foi medido ou não é comparado com zero, pois quando no comando enviado pelo Atmis o valor de um poluente vem vazio a base de dados interpreta esse vazio como sendo zero. Esta abordagem está sujeita a erros pois é possível que por vezes a leitura de um poluente seja um zero absoluto. No entanto, tendo em conta que as médias enviadas vêm com a precisão de 4 casas decimais a probabilidade de uma medição ser 0,0000 será muito reduzida.

À medida que é determinado o número de medições de um poluente é calculado também a sua soma. É agora possível iniciar o cálculo do índice. Este cálculo só ocorre se as cinco condições seguintes forem superadas:

```

if($indice_no2>=11){
    if(sizeof($o3_indice)>=11){
        if((sizeof($pm10_fundo)+sizeof($pm10_trf))>=11){
            if(sizeof($so2_indice)>=11){
                if(sizeof($co_indice)>=11){

```

A ordem pela qual as condições estão dispostas não é ao acaso. Existem poluentes obrigatórios no cálculo do índice de uma área e outros que devem ser só considerados caso sejam medidos nas estações dessa área. Os poluentes obrigatórios são o NO₂, o O₃ e as PM₁₀. No caso de Coimbra, como são medidos todos os poluentes, todos os eles devem entrar no cálculo do índice.

O próximo passo é o cálculo das médias de cada poluente tendo em conta as medições que entraram na contagem anterior. Esta média é calculada como o quociente da soma das medições pelo seu número:

```

$no2_media=$no2_media/$indice_no2;
$o3_media=$o3_media/sizeof($o3_indice);
$pm10_media=($pm10_fundo[0]+$pm10_fundo[0])/2;
$so2_media=$so2_media/sizeof($so2_indice);
$co_media=$co_media/sizeof($co_indice);

```

As médias assim determinadas são comparados com as gamas de concentrações de cada poluente, associadas a uma escala de cores. O índice varia de ‘Muito Bom’ a ‘Mau’ para cada poluente de acordo com a matriz de classificação seguidamente apresentada.

Poluente em causa / Classificação	CO		NO ₂		O ₃		PM ₁₀		SO ₂	
	Min	Máx	Min	Máx	Min	Máx	Min	Máx	Min	Máx
Mau	10 000	-----	400	-----	240	-----	120	-----	500	-----
Fraco	8 500	9999	200	399	180	239	50	119	350	499
Médio	7 000	8499	140	199	120	179	35	49	210	249
Bom	5 000	6999	100	139	60	119	20	34	140	219
Muito Bom	0	4999	0	99	0	59	0	19	0	139

Tabela 2 – Classificação do Índice de Qualidade do Ar proposto para o ano 2015, APA. Valores em $\mu\text{g}/\text{m}^3$.

Esta comparação é feita poluente a poluente, consoante os valores da tabela, da seguinte forma:

```
function indice_no2($no2_media){
  if ($no2_media >=0 && $no2_media<99){
    $indice_no2="GreenYellow";
  }
  elseif ($no2_media >=100 && $no2_media<139){
    $indice_no2="Green";
  }
  elseif ($no2_media >=140 && $no2_media<199){
    $indice_no2="Yellow";
  }
  elseif ($no2_media >=200 && $no2_media<399){
    $indice_no2="DarkOrange";
  }
  elseif ($no2_media >=400){
    $indice_no2="Red";
  }
  return $indice_no2;
}
$indiceno2=indice_no2($no2_media);
```

O índice geral é agora finalmente calculado com a função ‘indice_geral’ que procura a pior classificação de cada um dos poluentes:

```
function indice_geral($indicepm10, $indiceso2, $indiceco8h, $indiceno2, $indiceo3){
  if($indicepm10=="Red" || $indiceso2=="Red" || $indiceco8h=="Red" || $indiceno2=="Red" || $indiceo3=="Red"){
    $indice_geral="Red";
  }
  elseif($indicepm10=="DarkOrange" || $indiceso2=="DarkOrange" || $indiceco8h=="DarkOrange" || $indiceno2=="DarkOrange" || $indiceo3=="DarkOrange"){
    $indice_geral="DarkOrange";
  }
  elseif($indicepm10=="Yellow" || $indiceso2=="Yellow" || $indiceco8h=="Yellow" || $indiceno2=="Yellow" || $indiceo3=="Yellow"){
    $indice_geral="Yellow";
  }
  elseif($indicepm10=="Green" || $indiceso2=="Green" || $indiceco8h=="Green" || $indiceno2=="Green" || $indiceo3=="Green"){
    $indice_geral="Green";
  }
  else{
    $indice_geral="GreenYellow";
  }
  return $indice_geral;
}
$indicegeral=indice_geral($indicepm10, $indiceso2, $indiceco8h, $indiceno2, $indiceo3);
```

Todas as condições para o cálculo do índice geral, em especial o condicionamento do número mínimo de medições, implica que o índice só esteja disponível por volta das 15h de cada dia

(admitindo um atraso de 4 horas na entrega das medições das 11 horas da manhã). Até lá o índice assume a cor cinzenta. Além disso, no início deste último ficheiro é imposta uma condição que garante que a partir da meia-noite não é mostrado o índice do dia anterior. Este feito é conseguido recuperado a última data e hora da última entrada da tabela *qualidades_medias*: se este valor for inferior à data da meia-noite do dia atual o índice é imediatamente classificado como ‘cinzento’.

Mais informações relativas a este método de cálculo podem ser encontradas no *website* da APA no separador ‘Índice’.

Apêndice C – Textos Informativos

Neste apêndice estão reunidos todos os textos informativos que acompanham cada um dos painéis do Dashboard. Estas informações são exibidas ao clicar sobre qualquer um dos títulos desses painéis. Está também presente o conteúdo de “Acerca” deste projeto, cuja ligação está presente no canto inferior direito do Dashboard.

Modal Meteorologia

Os valores de meteorologia são fornecidos pela estação meteorológica do Departamento de Engenharia Mecânica da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Esta estação foi instalada em Dezembro de 2013 e veio substituir uma anterior com mais de uma década de existência.

Funcionamento da estação

Esta estação é do tipo *Davis VantagePro2 Plus* e os dados recolhidos são geridos através do software *weewx*. A estação recolhe a cada 5 minutos os mais variados índices e valores meteorológicos, entre os quais os mostrados neste painel: temperatura em graus Celcius, velocidade e direção do vento em metros por segundo (m/s) e graus respetivamente, a pluviosidade em milímetros por hora (mm/hora), pressão atmosférica em hectopascal (hPa) a humidade relativa em percentagem, a radiação solar em Watt por metro quadrado (W/m²) e o índice de UV.

O índice de UV é um indicador da intensidade da radiação na parte ultravioleta do espectro solar num determinado momento. Os raios ultravioleta desempenham um papel determinante em muitos processos na biosfera, com muitos efeitos benéficos, podendo no entanto causar graves prejuízos para a saúde se o nível de UV exceder certos limites. Na tabela seguinte encontram-se as recomendações de proteção para cada índice de UV.

Índice UV	Cor	Risco de danos provocados pela exposição ao sol sem proteção para o adulto médio	Proteção recomendada
0 - 2.9	Verde	"Baixo"	Usar óculos de sol em dias solarengos; Usar protetor solar se houver neve no chão ou se tiver pele clara.
3 - 5.9	Amarela	"Moderado"	Tenha precauções, tais como cobrir-se, se estiver na rua; Mantenha-se à sombra a meio do dia quando o sol é mais forte.
6 - 7.9	Laranja	"Alto"	Cubra o corpo com roupas que protejam do Sol, use protetor solar com SPF superior a 30, use chapéu, reduza a exposição ao solar no intervalo das 10h às 15h e use óculos de sol.
8 - 10.9	Vermelha	"Muito alto"	Usar protetor solar com SPF superior a 30, usar camisola, chapéu e óculos de sol. Não permanecer ao Sol durante muito tempo.
11 +	Violeta	"Extremo"	Tenha todas as precauções. Usar camisola de mangas compridas e calças, chapéu, óculos de sol, protetor solar com SPF superior a 30. Evitar o Sol nas horas de maior intensidade.

Tabela 3 – Escala de índice UV estabelecido pela EPA (United States Environmental Protection Agency).

Obtenção e frequência dos valores

Após cada aquisição de dados, estes são armazenados localmente na estação e enviados para um servidor que se encarrega de fornecer os valores necessários ao Smart City Coimbra Dashboard a cada 5 minutos. Após a leitura do feed pelo software do dashboard, os valores são guardados numa base de dados para que possam ser acedidos sempre que necessário.

É o feed do portal OpenweatherMap que fornece o estado do tempo, representado por um ícone no painel Meteorologia, uma vez que a estação meteorológica do DEM não tem a capacidade de analisar os valores medidos e fazer uma previsão do estado do tempo atual e/ou futuro.

Um utilizador que pretende obter mais informações pode aceder ao portal web WeatherUnderground, uma página *web* dedicada à partilha de informação meteorológica. A estação meteorológica do DEM é uma das estações usadas para recolha dos valores meteorológicos em Coimbra e, a partir desta informação, são apresentados as mais variadas previsões, gráficos e informações complementares.

Modal Nível do Rio

O nível do rio Mondego é fornecido pelo Sistema Nacional de Informação de Recursos Hídricos, o SNIRH. Este sistema de informação de recursos hídricos pertencente à Agência Portuguesa do Ambiente, I.P. (APA, I.P.) funciona desde 1995 e possui diversos serviços:

boletins mensais sobre recursos hídricos, disponibilização de séries temporais de dados hidrometeorológicos e de qualidade da água, disponibilização de informação técnica sobre recursos hídricos e disponibilização de informação Geográfica.

As duas estações usadas no dashboard, estação Ponte de Santa Clara e a estação Ponte Açude, são estações hidrométricas e o valor apresentado refere-se ao nível hidrométrico instantâneo em metros. A localização das estações está incluída no mapa do painel Locais de Interesse na categoria “Nível do Rio”.

Funcionamento das estações

Ambas as estações possuem teletransmissão e são constituídas por um sistema de armazenamento de dados (registador de dados), um sistema de alimentação (painel solar e bateria), um sistema de comunicação (modem), um sensor de pressão hidrostática e uma escala hidrométrica.

O sensor de pressão permite a medição do nível da água no rio e é aferido pela escala hidrométrica. A cota do zero da escala está referenciada ao nível médio do mar e o seu valor é de 15.45 m. Com a cota do zero da escala é possível obter a cota associada ao nível hidrométrico observado (soma dos dois valores).

Obtenção e frequência dos valores

Até ao momento da concretização desta versão do Dashboard não foi conseguida a ligação à base de dados do SNIRH e portanto os valores apresentados neste painel são fictícios.

Para aceder aos valores validados e outras informações sobre outras estações pode aceder ao portal do [SNIRH](#). O portal também divulga sínteses mensais temáticas que visam a caracterização das disponibilidades hídricas nacionais, relatórios técnicos, cartografia sobre recursos hídricos (por exemplo zonas inundáveis), documentos técnicos e fotografias relacionadas com os recursos hídricos.

Modal Qualidade do Ar

Os valores das concentrações de gases poluentes presentes neste painel são fornecidos pela Comissão de Coordenação e Desenvolvimento Regional do Centro e disponibilizados pela Agência Portuguesa do Ambiente.

As estações presentes no painel Qualidade do Ar, localizadas no Instituto Geofísico de Coimbra e na Avenida Fernão de Magalhães, são de dois tipos distintos. A primeira é do tipo Fundo e a segunda é do tipo Tráfego. A sua localização física pode ser consultada no painel Locais de Interesse na categoria Qualidade do Ar.

Funcionamento das estações

Os valores das concentrações dos cinco poluentes, Ozono, Dióxido de Azoto, Monóxido de Carbono, Dióxido de Enxofre e as partículas inaláveis ou finas, cujo diâmetro médio é inferior a 10 micra* (PM10), são medidos nas estações e posteriormente enviados para o servidor central da CCDR que os entrega à APA. Na APA é usado um sistema que faz a análise dos dados e os classifica como válidos ou não. Os valores inválidos são retificados.

(*) Micron - Um micron (representado pelo caractere grego μ) é a unidade métrica normalmente utilizada para definir o tamanho das partículas em suspensão no ar. Um micron é igual a 0,000001 m, isto é a milionésima parte do metro (mm).

Cálculo do Índice de Qualidade do Ar

O índice de qualidade do ar de uma determinada área resulta da média aritmética calculada para cada um dos poluentes medidos em todas as estações da rede dessa área. O tipo de média varia entre cada poluente:

- Dióxido de azoto (NO₂) - médias horárias
- Dióxido de enxofre (SO₂) - médias horárias
- Ozono (O₃) - médias horárias
- Monóxido de carbono (CO) - médias de 8 horas consecutivas
- Partículas inaláveis (PM₁₀) - média diária

Os valores assim determinados são comparados com as gamas de concentrações associadas a uma escala de cores. O índice varia de Muito Bom a Mau para cada poluente de acordo com a matriz de classificação seguidamente apresentada.

Poluente em causa / Classificação	CO		NO2		O3		PM10		SO2	
	Min	Máx	Min	Máx	Min	Máx	Min	Máx	Min	Máx
Mau	10 000	-----	400	-----	240	-----	120	-----	500	-----
Fraco	8 500	9999	200	399	180	239	50	119	350	499
Médio	7 000	8499	140	199	120	179	35	49	210	249
Bom	5 000	6999	100	139	60	119	20	34	140	219
Muito Bom	0	4999	0	99	0	59	0	19	0	139

Tabela 4 – Classificação do Índice de Qualidade do Ar proposto para o ano 2015, APA. Valores em $\mu\text{g}/\text{m}^3$.

Independentemente de quaisquer fatores de interação entre diferentes poluentes, o grau de degradação da qualidade do ar está dependente da pior classificação verificada entre os diferentes poluentes considerados, pelo que o IQar é definido a partir do poluente que apresenta pior classificação (ex: valores médios registados numa dada área: SO₂ - 35 $\mu\text{g}/\text{m}^3$ (Muito Bom), NO₂ - 180 $\mu\text{g}/\text{m}^3$ (Médio); CO - 6000 $\mu\text{g}/\text{m}^3$ (Bom), PM₁₀ - 15 $\mu\text{g}/\text{m}^3$ (Muito Bom) e O₃ - 365 $\mu\text{g}/\text{m}^3$ (Mau) o IQar é mau, devido às concentrações observadas para o ozono).

Precauções em função da Qualidade do Ar

O ar que respiramos afeta a saúde das populações, principalmente a dos grupos sensíveis cuja capacidade respiratória se encontra em formação ou debilitada (crianças, idosos e doentes asmáticos). No quadro seguinte apresentam-se alguns conselhos de saúde em função do índice IQar, mencionando também as condições meteorológicas normalmente associadas.

Índice	Tempo	Conselhos de Saúde
Mau	<ul style="list-style-type: none">- Anticiclone com vento fraco;- Estabilidade prolongada;- Depressão do Norte de África com uma corrente de SE no continente transportando poeiras do deserto;- Ozono: forte radiação / tempo quente contínuo.	Todos os adultos devem evitar esforços físicos ao ar livre. Os grupos sensíveis (crianças, idosos e indivíduos com problemas respiratórios) deverão permanecer em casa com as janelas fechadas e utilizando de preferência sistemas apropriados de circulação/ refrigeração de ar.
Fraco	<ul style="list-style-type: none">- Anticiclone com vento fraco;- Situações de transição do estado do tempo;- Estabilidade;- Depressão do Norte de África com uma corrente de SE no continente transportando poeiras do deserto;- Ozono: forte radiação / temperaturas elevadas associadas a dias de céu limpo.	As pessoas sensíveis (crianças, idosos e indivíduos com problemas respiratórios) devem evitar atividade físicas intensas ao ar livre. Os doentes do foro respiratório e cardiovascular devem ainda respeitar escrupulosamente os tratamentos médicos em curso ou recorrer a cuidados médicos extra, em caso de agravamento de sintomas. A população em geral deve evitar a exposição a outros fatores de risco, tais como fumo do tabaco e a exposição a produtos irritantes contendo solventes na sua composição.
Médio	<ul style="list-style-type: none">- Diversas situações meteorológicas com características de tempo agradáveis.	As pessoas muito sensíveis, nomeadamente crianças e idosos com doenças respiratórias devem limitar as atividades ao ar livre.
Bom	<ul style="list-style-type: none">- Passagem de frentes com atividade moderada;- Outras situações meteorológicas com ventos moderados.	Nenhuns.
Muito Bom	<ul style="list-style-type: none">- Vento moderado a forte;- Temperaturas frescas;- Ocorrência de precipitação;- Passagem de frentes com atividade moderada.	Nenhuns.

Tabela 5 - Conselhos de saúde em função do índice IQar, APA.

Obtenção e frequência dos valores

Em cada estação existe um *modem* analógico que envia, em resposta a uma interrogação, os valores medidos continuamente para o servidor central da CCDRC, onde são guardados numa base de dados. Estas medições são posteriormente enviadas, com recurso ao sistema Atmis, para o servidor da APA e para o servidor do Dashboard em simultâneo. Na APA os valores são revistos e, caso se encontrem anomalias, estas são reportadas à CCDRC que confirma os valores ou envia novas medições. Os valores já validados são depois usados no sistema *Qualar* na APA para determinar o índice de qualidade do ar.

Todos os valores enviados para o servidor do Dashboard são guardados numa base de dados para que sejam acedidos pelo Dashboard sempre que necessário. No momento de envio dos valores estes ainda não foram validados, pelo que é possível que os valores mostrados neste

painel não correspondam à realidade. Para consultar estes e outros valores pode-se aceder ao portal Qualar da APA.

Modal Energia Elétrica

A energia elétrica consumida em Coimbra é fornecida por três subestações: Alegria, Relvinha e Alto de S. João. A subestação da Alegria tem uma potência instalada de 60 MVA, a subestação da Relvinha tem uma potência instalada de 80 MVA e a subestação do Alto de S. João tem uma potência instalada de 40 MVA. Os valores dos consumos serão facultados pela “EDP Distribuição” mas à data da construção deste protótipo dashboard a ligação à base de dados da EDP não foi conseguida. Por essa razão foram usados os valores de consumo de energia elétrica do Departamento de Engenharia Eletrotécnica e de Computadores da Universidade de Coimbra, cedidos pela empresa Streamline, e obtidos de uma base de dados.

Modal Água

A gestão do abastecimento de água a vários concelhos da Região Centro é feita pelas Águas do Mondego. Devido a limitações técnicas no acesso ao valor de caudal de água consumido e dos respetivos gastos energéticos associados por parte do seu sistema de controlo de dados, até ao momento da conclusão desta versão do Dashboard ainda não havia uma solução para a melhor forma de obter os dados. Por esse motivo, são exibidos no gráfico deste painel os valores de consumo de água do Departamento de Engenharia Eletrotécnica e de Computadores da Universidade de Coimbra, cedidos pela empresa Streamline, e obtidos de uma base de dados.

Modal Gás

A Lusitaniagás é a empresa concessionária para a distribuição de gás na região Litoral Centro, do qual faz parte a cidade de Coimbra. Essa distribuição é feita em 6 PRM (pontos de regulação e medição) localizados dentro da cidade e arredores: Condeixa, Cernache, Casa do Sal, Oliveira do Hospital, Loreto, e Santa Clara.

Obtenção e frequência dos valores

O volume de gás consumido, em metros cúbicos, é medido uma vez por dia nos 6 PRM. Estes valores são cedidos pela empresa por envio por FTP (*File Transfer Protocol*) de um ficheiro Excel, que é atualizado diariamente entre as 7h00 e as 11h00 com os valores de consumo de gás em todos os pontos mencionados, entre as 00h00 e as 23h59 do dia anterior.

Os valores são lidos e armazenados numa base de dados local sendo depois acedidos pelo Dashboard para criar o gráfico de consumo de gás. Este gráfico é atualizado uma vez por dia, uma vez que a frequência dos valores recebidos é diária.

Modal Consumo de Recursos

Neste painel encontram-se reunidos alguns indicadores relativos aos consumos da cidade: energia elétrica, gás e água. Devido à diferente natureza destes consumos foram escolhidos diferentes indicadores para cada um.

Energia Elétrica

- Pontas do dia atual: valor máximo e mínimo;
- Potência média: definida como o quociente entre a energia gasta num período de tempo e a duração desse mesmo período. A potência média é atualizada a cada 15 minutos e refere-se ao consumo até ao momento do dia atual;
- Fator de carga: índice que ilustra a eficiência da utilização da infraestrutura elétrica, é obtido pela relação entre a potência média e a potência máxima, durante um período de tempo. Esse período de tempo corresponde ao tempo decorrido até ao momento do dia atual;

Água

- Pontas do dia atual: valor máximo e mínimo;
- Valor médio: valores médios de consumo de dia atual até ao momento e do dia anterior no período homólogo;
- Comparação: é feita a comparação entre o consumo de água do dia atual até ao momento com o necessário encher uma piscina com um volume de 80m^3 .

Gás

- Máximos: valor máximo desta semana e valor máximo da semana anterior;
- Valor médio: valores médios de consumo da semana atual até ao momento e da semana anterior no período homólogo;
- Comparação: conversão dos m^3 consumidos para MWh;

Modal Produção de Resíduos

O painel Produção de Resíduos mostra valores estáticos correspondentes aos resíduos produzidos e tratados no concelho de Coimbra nos anos de 2011,2012 e 2013 e foram cedidos pela Divisão do Ambiente do Departamento de Desenvolvimento Social e Ambiente da Câmara Municipal de Coimbra. Por serem estáticos, novos valores serão acrescentados apenas quando os mesmos estiverem disponíveis.

Os resíduos do concelho de Coimbra são recolhidos pela Câmara Municipal e prestadores de serviço concessionados. A recolha é feita maioritariamente em contentores e ecopontos embora resíduos de natureza especial tais como os bio-resíduos e os resíduos de grandes produtores sejam recolhidos na fonte.

Infraestruturas	2011	2012	2013
Ecopontos (Nº)	473	474	484
Ecopontos Subterrâneos	---	---	472
Ecopontos de superfície	---	---	12
Contentores (Nº)	5 636	6 014	5 939
100L	287	754	580
800L	5 318	5 230	5 324
3 000L	7	11	11
5 000L	14	18	24
Capacidade Instalada (m3)	4 290	4 389	4 476
100L	34	82	64
800L	4.254	4.184	4.259
3 000L	21	33	33
5 000L	70	90	120
Oleões (Nº)	11	24	24

Tabela 6– Infraestruturas do Município de Coimbra

As viaturas de recolha dos resíduos, após terminarem o seu trajeto, dirigem-se ao Centro Integrado de Tratamento e Valorização de Resíduos Sólidos Urbanos, CITVRSU, em Vil de Matos onde são pesados e depositados os seguintes resíduos:

- RSU
- MONOs
- REEE
- Papel/cartão
- Embalagens
- Vidro

Os primeiros dois tipos de resíduos são considerados “Não Valorizados” e têm como destino o aterro sanitário. Ainda assim, alguns RSU, após uma triagem, são reaproveitados para tratamento mecânico-biológico (compostagem). Os restantes resíduos são considerados “Valorizáveis” e encaminhados para as estações de triagem e depois para as indústrias recicladoras.

Os resíduos não incluídos na lista são transportados para operadores de gestão de resíduos licenciados pelo Ministério do Ambiente, Ordenamento do Território e Energia (MAOTE).

No processo de recolha e transporte, as viaturas emitem gases poluentes para a atmosfera, nomeadamente o dióxido de carbono. Este valor é estimado pela Divisão de Ambiente com base nos litros de combustível consumidos por cada viatura.

	2011	2012	2013
Emissões de CO₂ (kg CO₂)	1.069.085	1.135.732	910.718

Tabela 7 – Emissões de CO₂ das viaturas de recolha de resíduos

Modal Emissões de CO₂

O painel Emissões de CO₂ destina-se, como o nome indica, a mostrar as emissões, estimadas, de dióxido de carbono resultantes da produção de energia elétrica e do abastecimento de água da cidade.

Para estimar estas emissões foi necessário conhecer os valores de repartição de produção de energia elétrica em Portugal e ainda os fatores de emissão de CO₂ (g/kWh). Segundo os Dados Técnicos, publicado pela REN com divulgação dos principais indicadores estatísticos dos setores da sua atividade em 2013, 22% e 14% da produção total de energia elétrica correspondem à queima de carvão e de gás natural. Os fatores de emissão utilizados foram os divulgados no documento “Recomendação sobre Rotulagem de Energia Elétrica” em Dezembro de 2011, disponibilizado pela ERSE. Segundo este documento, para o carvão o fator é de 973 g/kWh e para o gás natural de 350 g/kWh.

Cálculo das Emissões

As emissões de CO₂ são determinadas pela multiplicação dos valores da produção de energia elétrica pelos fatores de emissão, para cada uma das fontes de energia, conforme a expressão seguinte:

$$Emissões\ de\ CO_2 = Energia\ elétrica\ produzida * Fator\ de\ emissão\ de\ CO_2 [g]$$

Com esta expressão, e usando os valores de potência elétrica de consumo que se encontram guardados na base de dados (a cada 15 minutos), obtemos as expressões abaixo.

$$\text{Emissões de } CO_2|\text{Carvão} = (\text{Potência de consumo} * 0,25\text{horas}) * 0,22 * 973[\text{g}]$$

$$\text{Emissões de } CO_2|\text{Gás Natural} = (\text{Potência de consumo} * 0,25\text{horas}) * 0,14 * 350 [\text{g}]$$

As emissões de dióxido de carbono relativas ao consumo de energia elétrica são estimadas com base nos valores médios de potência elétrica de consumo recolhidos e mostrados no painel Energia, a cada quarto de hora. Já as emissões relacionadas com o consumo de água são obtidas com o valor consumo de energia elétrica realizado no abastecimento à cidade (bombagem, filtragem, etc..), pelo que este valor já se encontra incluído no total de emissões de dióxido de carbono resultantes do consumo de energia elétrica da cidade. Este último valor seria fornecido pelas Águas do Mondego. Uma vez que até ao momento não foi conseguida a ligação à sua base de dados, são usados valores fictícios.

Recomendações

Com esta informação queremos consciencializar os consumidores de energia elétrica das consequências do seu consumo e convidá-los a fazerem um uso da mesma cada vez mais consciente e responsável. Desta forma, estarão a defender não só o ambiente e a sustentabilidade do planeta, como também a poupar significativamente na fatura da eletricidade.

Recomendamos os seguintes gestos simples para diminuir o consumo de eletricidade:

- Substitua as lâmpadas incandescentes ou halogéneas por lâmpadas economizadoras;
- Evite ter as luzes ou os equipamentos ligados, quando não for necessário;
- Ao comprar um novo equipamento, verifique a etiqueta energética e opte por aquele que apresenta menor consumo de energia;
- Não deixe os seus aparelhos em modo *stand-by*, desligue-os no botão;
- Retire os carregadores de bateria da ficha imediatamente após o seu aparelho estar carregado.

É a responsabilidade individual no consumo que torna possível gerir as fontes usadas na produção de energia elétrica, na medida em que a redução da produção de energia elétrica permite não só reduzir diretamente as emissões de dióxido de carbono como também dar maior relevância à utilização de energias renováveis em detrimento do consumo de combustíveis fósseis, elementos poluidores da atmosfera.

Para diminuir a fatura da água recomendamos as seguintes práticas:

- Mantenha as torneiras sempre bem fechadas e verifique regularmente se não há fugas;
- Utilize a água racionalmente para lavar roupas, louças, limpeza e banho;

Modal Eventos

A Agenda 7-Coimbra é uma plataforma *online* dedicada à divulgação dos eventos culturais na cidade de Coimbra. Este projeto, que era inicialmente da Universidade de Coimbra, faz parceira a Câmara Municipal de Coimbra desde 2013 e conta já com mais de 60 entidades que contribuem para a manter atualizada e funcional, o que faz dela uma referência nos canais de informação e de promoção de eventos culturais.

Nesta plataforma encontrará exposições, teatro, música, cinema, conferências, desporto e demais atividades que todos os dias acontecem em Coimbra.

Obtenção e frequência dos valores

A Agenda 7-Coimbra é disponibilizada sob um formato concebido especialmente para o Dashboard. Por omissão o período de tempo exibido é o mês atual, mas é possível consultar outros meses, anteriores ou futuros.

Para mais informações sobre esta plataforma e mais detalhes dos eventos realizados em Coimbra aceda ao portal [Agenda 7-Coimbra](#).

Modal Locais de Interesse

Com o intuito de promover e dar a conhecer os locais mais importantes da cidade de Coimbra, foi elaborado um mapa com os pontos considerados mais relevantes. Este mapa foi desenvolvido com base no Google Maps Engine.

Os pontos selecionados foram divididos em 6 categorias, cada uma delas associada a uma cor diferente: Jardins (verde), Museus e Monumentos (bordô), Energia (amarelo), Qualidade do Ar (azul claro), Meteorologia (castanho) e Água (azul escuro).

Jardins

A esta categoria foi atribuída a cor verde e nela foram incluídas todas as áreas verdes relevantes da cidade, quer em tamanho quer em história. As 11 áreas verdes escolhidas são o Jardim Botânico da Universidade de Coimbra, o Jardim da Manga, o Jardim da Sereia, o Penedo da Saudade, os Jardins da Quinta das Lágrimas, a Mata Nacional do Choupal, o Parque Verde do Mondego, o Parque do Choupalinho, o Jardim dos Montes Claros, o Jardim da Praça dos Heróis do Ultramar e o Jardim da Avenida Sá da Bandeira. Todos estes espaços verdes são de entrada livre à exceção dos Jardins da Quinta das lágrimas.

Museus e Monumentos

Nesta categoria, associada à cor bordô, estão reunidos os principais edifícios de relevo na cidade que, pela sua idade, história ou oferta cultural e científica, são pontos de passagem (quase) obrigatória. Estes 21 pontos de interesse são o Museu Nacional Machado de Castro, o Museu da Ciência, o Exploratório Infante Dom Henrique, o Museu dos Transportes urbanos de Coimbra, o Museu da Água de Coimbra, o Museu da Cidade de Coimbra, a Casa-Museu Miguel Torga, a Sé Velha e a Sé Nova de Coimbra, a Universidade de Coimbra e a Biblioteca Joanina, o Arco de Almedina, a Igreja e Mosteiro de Santa Cruz, o Mosteiro de Santa Clara-a-Velha, a Igreja Rainha Santa, a Igreja de Santo António dos Olivais, o Pavilhão Centro de Portugal, o Portugal dos Pequenitos, a Casa Museu Bissaya Barreto, o Memorial da Irmã Lúcia e o Pátio da Inquisição.

Para mais informações detalhadas sobre estes e outros locais históricos ou pontos turísticos na cidade, consulte o portal do [Turismo de Coimbra](#) onde irá encontrar também horários de abertura e preços.

Energia

Tanto esta como as próximas categorias foram criadas para indicar geograficamente os pontos onde é feita a recolha de alguns dos valores mostrados no dashboard. Nesta categoria estão incluídas as subestações elétricas que abastecem a cidade de Coimbra e das quais são recolhidos os valores de energia consumida. Essas subestações são 3 e são denominadas Subestação da Alegria, Subestação do Alto de São João e Subestação da Relvinha.

Qualidade do Ar

À semelhança da categoria anterior, nesta categoria encontram-se indicados os locais onde são medidos os vários poluentes atmosféricos. Estes dois locais encontram-se em zonas diferentes por terem propósitos diferentes. A primeira estação é uma estação de tráfego, e tal como nome indica encontra-se numa zona de elevado tráfego automóvel, na Avenida Fernão de Magalhães. A segunda estação, sendo uma estação de fundo, encontra-se numa zona teoricamente menos sujeita a poluição por tráfego, no Instituto Geofísico.

Meteorologia

Nesta categoria é indicado a localização da estação responsável pela recolha dos valores meteorológicos na cidade. Esta estação está situada no Departamento de Engenharia Mecânica da Universidade de Coimbra.

Água

Esta categoria é destinada aos locais de onde se situam as estações que abastecem a cidade. No entanto, apenas a subestação da Boavista está indicada.

Modal Notícias

Neste painel são exibidas as mais recentes notícias na cidade de Coimbra. Estas provêm do website “Destakes” e do jornal online “Notícias de Coimbra”, sendo a ordem das notícias ditada pela sua data (das mais recentes às mais antigas).

Modal Farmácias

A informação relativa às farmácias em serviço é fornecida pelo portal [farmácias de serviço.net](http://farmacias.de.servico.net). O portal surgiu em 2006 com o intuito de informar os cidadãos de forma fácil e útil do horário de funcionamento das farmácias.

Modal Acerca

A Universidade de Coimbra (UC) apresenta o Smart Coimbra Dashboard, plataforma online que apresenta informação essencial relativa à cidade de Coimbra de forma atualizada e concentrada num único sítio.

Nascido no âmbito do projeto “SmartCoimbra”, sustentado no protocolo existente entre a UC e a Câmara Municipal de Coimbra, e no qual Coimbra é considerada com potencial para ser uma *smart city*, o Smart Coimbra Dashboard foi concebido para dar alguma visibilidade pública de dados sobre o funcionamento global da cidade.

O objetivo a atingir é fazer do Smart Coimbra Dashboard uma referência pública de dados do funcionamento da cidade de Coimbra.

Este painel de informações é a segunda versão de um protótipo e deve ser considerado como tendo qualidade "alpha".

Agradecemos aos seguintes fornecedores de dados, por proverem as informações necessárias usadas neste projeto:

- Departamento de Engenharia Mecânica
- Sistema Nacional de Informação de Recursos Hídricos
- Comissão de Coordenação e Desenvolvimento Regional do Centro
- Agência Portuguesa do Ambiente
- EDP Distribuição
- Águas do Mondego

- Lusitaniagás
- Divisão do Ambiente do Departamento de Desenvolvimento Social e Ambiente da Câmara Municipal de Coimbra
- SMTUC
- Agenda7
- Google Maps Engine
- Notícias de Coimbra
- Destakes
- Farmácias de Serviço
- Streamline

Anexo A – Protocolo e Anexo Técnico

Protocolo da UC com a Entidade sobre a obtenção de dados para o Dashboard de Coimbra

PROTOCOLO

Considerando:

Que a Universidade de Coimbra tem em curso o desenvolvimento de um projeto de um painel de pilotagem, ou *dashboard*, da cidade de Coimbra que tem como objetivo criar uma instância na Internet de um painel de indicadores e informações sobre a cidade, usual em projetos de cidade inteligente, ou *Smart City*, tendo em vista:

- a) caracterizar a utilização de recursos pela cidade (energia sob diversas formas, água);
- b) caracterizar os impactos provocados pela cidade no ambiente (resíduos, emissões) e a qualidade do ar no ambiente urbano;
- c) concentrar num único endereço um conjunto de informações úteis aos cidadãos residentes ou visitantes (transportes, meteorologia, informação turística, notícias, etc.) evitando dispersão por diversas plataformas para a obtenção de informações práticas mais correntes;

Que este *dashboard* permite que a cidade não só se revele, como também permite mais facilmente promover e organizar a sensibilização dos cidadãos para o uso sustentável dos recursos em meio urbano;

Que o efeito de um dispositivo deste tipo é tanto maior quanto mais dinâmico for na caracterização da realidade, sendo, assim, recomendável que a informação a apresentar tenha uma frequência de atualização elevada;

Que a Nome da Entidade reconhece a importância e utilidade pública de um instrumento deste tipo na sensibilização dos cidadãos para as questões da sustentabilidade e na evolução das realidades urbanas para o paradigma das cidades inteligentes e amigáveis, contribuindo também, assim, para o progresso e para o desenvolvimento responsável da Região Centro,

A **Universidade de Coimbra**, pessoa coletiva de direito público nº 501 617 582, com sede no Paço da Escolas, em Coimbra, neste ato representada por Nome do Representante, e de ora em diante designada abreviadamente por UC,

e

a **Nome da Entidade**, organismo desconcentrado da Presidência do Conselho de Ministros, com tutela conjunta com o Ministério A e sede na Rua Nome, Número, em Cidade, neste ato representada por Nome do Representante, e de ora em diante designada abreviadamente por Entidade,

estabelecem entre si o presente protocolo que se regerá de harmonia com as cláusulas seguintes:

Cláusula 1ª

(Objecto)

O presente Protocolo visa estabelecer as relações entre as partes signatárias, no quadro do funcionamento de um painel de pilotagem da cidade de Coimbra, doravante designado “dashboard”, o qual visa principalmente informar sobre a utilização de recursos pela cidade e sobre os correspondentes impactos ambientais, contribuindo para sensibilizar os cidadãos para as questões da sustentabilidade na realidade urbana.

Cláusula 2ª

(Compromissos da UC)

- 1 – O dashboard proporciona aos utilizadores a identificação da fonte dos dados com garantia de interatividade intuitiva e amigável.
- 2 – O método de tratamento dos dados recebidos da Entidade é previamente validado com a Entidade. A validação do método é efetuada através de, pelo menos, mensagem de correio eletrónico entre interlocutores aceites reciprocamente como adequados para estabelecer esta validação.
- 2 – O dashboard proporciona aos utilizadores informação suficiente para uma correta interpretação de todos os atributos relevantes dos dados disponibilizados, bem como informação sobre as metodologias de cálculo dos valores apresentados.
- 3 – É atualizada tão rapidamente quanto possível a forma de apresentação da informação sempre que tal seja necessário para uma correta adequação a possíveis alterações que o justifiquem, ou por indicação da Entidade ou por deteção por parte da UC daquela eventual necessidade.
- 4 - A UC compromete-se a informar a Entidade sobre qualquer alteração de natureza técnica no suporte ao dashboard que tenha influência na comunicação de dados.

Cláusula 2ª

(Margem de atuação autónoma da UC)

- 1 - A UC altera a disposição dos elementos constantes do dashboard e os grafismos sempre que considerar oportuno ou necessário.
- 2 - A UC processa os dados da forma que for mais adequada e mais útil à disseminação de informação relevante para os cidadãos, de acordo com método validado pela Entidade.
- 3 - Sempre que necessário ao aperfeiçoamento/reestruturação da informação disponibilizada ou da disposição dos elementos gráficos, a UC pode solicitar apoio/colaboração à Entidade.

Cláusula 3ª

(Compromissos da Entidade)

- 1 - A Entidade garante o fornecimento dos dados sobre a qualidade do ar necessários à respetiva disponibilização no dashboard, de acordo com o estipulado no Anexo Técnico.
- 2 - A Entidade comunica à UC atempadamente qualquer alteração relativa às circunstâncias de produção da informação relevante, ao formato dos dados ou às características da comunicação, que possam ter influência na correta disponibilização de informação no dashboard.
- 3 - A Entidade comunica à UC qualquer alteração que se prenda com a identificação dos locais de origem dos dados ou com a necessidade de alterar/atualizar a informação contextual existente no dashboard.
- 4 - O Anexo Técnico referido no número 1 pode ser substituído sem necessidade de revalidação do presente protocolo, devido a alterações de natureza técnica no processo de disponibilização de dados pela Entidade à UC, podendo cada nova versão ser assinada por representantes da Entidade e da UC mutuamente acordados por comunicação corrente, designadamente por correio eletrónico.

Cláusula 4ª

(Direitos Entidade)

- 1 - Após comunicação à UC de alguma necessidade de alterar informação de contexto relativa a qualquer característica relevante dos dados disponibilizados, a Entidade deve ver rapidamente concretizada a alteração requerida.
- 2 - A Entidade tem a identificação institucional evidenciada no dashboard na sua qualidade de fornecedora de dados.
- 3 - A Entidade pode sugerir à UC alterações ou formas novas de proceder ao cálculo e disponibilização de informação/indicadores ao público, tendo em vista a melhoria da qualidade da informação disponibilizada.

Cláusula 5ª

(Compromisso comum)

As partes comprometem-se a efetuar uma pesquisa permanente de formas mais ágeis e eficientes de fazer circular os dados relevantes para alimentar o dashboard.

Cláusula 6ª

(Entrada em vigor e validade)

O presente protocolo entra em vigor na data da sua assinatura e é válido por tempo indeterminado.

Cláusula 6ª

(Modificação e cessação)

1 - Quaisquer alterações ao presente protocolo, designadamente o aumento do número de outorgantes, só serão válidas se constarem de documento escrito assinado pelos representantes das partes. Tal documento passa a constituir adenda ao protocolo, dele fazendo parte integrante.

2 - As partes poderão fazer cessar o presente protocolo por mútuo consentimento ou através de informação enviada pela parte interessada na cessação à outra parte, através de ofício assinado por representante habilitado, com sessenta dias de antecedência.

Pela Universidade de Coimbra

(assinatura)

Pelo Nome da Entidade

(assinatura)

ANEXO TÉCNICO

O presente Anexo Técnico está ligado ao Protocolo celebrado entre a Universidade de Coimbra (UC) e a Nome da Entidade (Entidade), no âmbito do projeto de *dashboard* da cidade de Coimbra.

Neste documento identificam-se, em x pontos, as condições técnicas em que se verifica a comunicação entre os sistemas que as duas entidades mobilizam para este projeto.

1. A Entidade tem a seu cargo estações de aquisição de dados na cidade Cidade, situadas respetivamente na Rua A e Rua B, as quais são interrogadas periodicamente por um sistema de monitorização.
2. Os dados recolhidos por estas estações são validados pelos serviços da Entidade e disponibilizados à UC através o sistema Sistema.
3. O envio dos dados é efetuado com o método Método.
4. A UC dá a conhecer à Entidade o identificador DNS ou o IP do servidor da UC que a Entidade parametriza no Sistema para o envio de dados.
5. As especificações do conteúdo das mensagens de dados incluídas nos comandos Comandos, bem como das mensagens de resposta, seguem de forma exata as especificações correspondentes do Sistema.