



Luís Miguel Calisto Galego

Licenciatura em Engenharia Eletrotécnica e de Computadores

**Desenvolvimento de uma Plataforma Web para a
disseminação de conhecimento no âmbito do Projeto
IPSTERS**

Dissertação para obtenção de grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Prof. José Fonseca, FCT-UNL

Coorientador: Prof. Tiago Cardoso, FCT-UNL



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2019

Desenvolvimento de uma Plataforma Web para a disseminação de conhecimento no âmbito do Projeto IPSTERS

Copyright © Luís Miguel Calisto Galego, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Em primeiro lugar, um agradecimento á minha família, em especial aos meus pais e á minha irmã por terem acreditado sempre em mim, por estarem sempre disponíveis em todos os momentos ao longo destes 5 anos e por me terem disponibilizado todas as ferramentas que tornaram este percurso possível.

Agradeço ao meu orientador, Professor José Fonseca, e ao meu coorientador, Professor Tiago Cardoso por se terem mostrado sempre disponíveis para ajudar com novas ideias e soluções para os todos os problemas que foram aparecendo ao longo deste percurso.

Agradeço também a todos os meus colegas do grupo CA3, por me terem acolhido neste grande grupo de investigação. Ao Leonardo um agradecimento especial pelo apoio, dedicação e ajuda durante a realização desta dissertação.

Agradeço a todos os meus amigos, pelo apoio que me deram durante estes 5 anos, aos amigos da universidade, espero ter sempre presente as memórias que fizemos durante este percurso.

RESUMO

O projeto IPSTERS está a ser desenvolvido pela Direção Geral de Território (DGT) em parceria com o UNINOVA e com a Nova Information Management Center (Magic center). Este projeto tem como objetivo principal a exploração das possíveis aplicações e avaliações das limitações do uso de algoritmos de inteligência artificial para realizar o tratamento da informação proveniente da infraestrutura IPSentinel.

Esta dissertação encontra-se enquadrada na tarefa de disseminação e comunicação da tecnologia do projeto IPSTERS, tendo como principal objetivo o desenvolvimento de uma plataforma web onde podem ser analisados dados tanto dos mapas de incêndios do ICNF como da interseção dos dados destes mapas com a carta de uso e ocupação de solo (COS) de Portugal.

O desenvolvimento desta plataforma foi realizado recorrendo á web framework Django, uma framework baseada em Python, e a várias aplicações suplementares desta mesma framework. Deste modo, a plataforma desenvolvida oferece dois serviços distintos, um serviço para a análise da informação dos incêndios onde pode ser analisado uma variedade de gráficos sobre a interseção dos dados dos incêndios e da COS, e um serviço para a visualização dos mapas, onde as áreas afetadas pelos incêndios dos anos de 2014-2018 podem ser analisadas com base num mapa interativo.

Palavras chave: IPSTERS, COS, mapas de incêndios ICNF, Python, Django

Abstract

IPSTERS is a project developed by Direção Geral de Território (DGT) in a partnership with UNINOVA and the Nova Information Management Center (Magic center). The main objective of this project is to explore possible applications and evaluate the restrictions on the use of artificial intelligence algorithms to process the information proven by the IPSentinel infrastructure.

This dissertation is framed in the task of technology dissemination and communication of the IPSTERS project, for this reason this dissertation focused on the development of a web platform where data can be analyzed from both the ICNF fire maps and the intersection of these data with the COS.

The development of this platform was done with the help of Django framework, a Python-based framework, and several supplemental applications of the same framework. This platform offers two kinds of services, the service responsible for the fire information analysis, where a variety of graphs where data from the intersection of burned and COS data can be analyzed, and the service responsible for the visualization of the maps where it's possible to analyze a map with a layer of areas affected by the fires from 2014 to 2018.

Keywords: IPSTERS, COS, ICNF Fire Maps, Python, Django

Índice

INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO.....	1
1.2. OBJETIVOS.....	2
ESTADO DE ARTE	5
2.1. FIRE INFORMATION FOR RESOURCE MANAGEMENT SYSTEM (FIRMS).....	5
2.2. COPERNICUS.....	6
2.2.1. Serviço de gerenciamento de emergências do Copernicus.....	8
2.3. FOGOS.PT.....	11
METODOLOGIA	13
3.1. PYTHON.....	13
3.1.1. Breve história do Python.....	14
3.2. DJANGO.....	14
3.2.1 Breve história do Django.....	15
3.2.2 Filosofias do Django.....	16
3.2.3 Graphos.....	18
3.2.4 Django Leaflet.....	18
3.2.5 GeoDjango.....	19
3.3. COS.....	23
3.4. ICNF.....	25
DESENVOLVIMENTO	27
4.1. TRATAMENTO DE DADOS.....	27
4.2. DESENVOLVIMENTO DO SERVIÇO PARA ANÁLISE DA INFORMAÇÃO DOS INCÊNDIOS.....	33
4.2.1. Definições do Django.....	33
4.2.2. Modelos do Django.....	34
4.2.3. Desenvolvimento da front-end.....	36
4.3. DESENVOLVIMENTO DO SERVIÇO PARA VISUALIZAÇÃO DOS MAPAS.....	39
4.3.1. Configuração do Django e da base de dados.....	39
4.3.2. Modelos GeoDjango.....	40
4.3.3. Desenvolvimento da front-end.....	41
RESULTADOS	45
5.1. INFORMAÇÃO DOS INCÊNDIOS.....	45
5.2. SERVIÇO DE VISUALIZAÇÃO DOS MAPAS DOS INCÊNDIOS.....	53
CONCLUSÕES	59
6.1. OBJETIVOS CUMPRIDOS.....	59

6.2. TRABALHO FUTURO.....	60
---------------------------	----

REFERÊNCIAS

ERRO! MARCADOR NÃO DEFINIDO.

ANEXO I	69
----------------	-----------

ANEXO II	71
-----------------	-----------

Lista de Figuras

FIGURA 1 - MAPA DAS ÁREAS ARDIDAS DE 2017 (FIRMS) [70]	6
FIGURA 2 - REPRESENTAÇÃO DO SATÉLITE SENTINEL 2. RETIRADO DE [71]	7
FIGURA 3 – INTERFACE DO EFFIS – APLICAÇÃO DO CURRENT SITUATION - FIRE DANGER. RETIRADO DE FORECAST [75]	10
FIGURA 4 – INTERFACE DO EFFIS – APLICAÇÃO DO <i>FIRE NEWS</i> . RETIRADO DE [74]	11
FIGURA 5 – INTERFACE DO <i>WEBSITE</i> FOGOS.PT - DETALHES DE UM INCÊNDIO. RETIRADO DE [72]	12
FIGURA 6 – INTERFACE DO <i>WEBSITE</i> FOGOS.PT – DETALHES DA ESTATÍSTICA DO DIA 07 DE AGOSTO DE 2019. RETIRADO DE [76]	12
FIGURA 7 - PÁGINA INICIAL DO DJANGO,.....	15
FIGURA 8 - EXEMPLOS DE GRÁFICOS GERADOS PELO GRAPHOS	18
FIGURA 9 - VISTA DO MAPA DO CAMPUS DA FCT-UNL USANDO LEAFLET	19
FIGURA 10 - SITE DE ADMINISTRADOR DO DJANGO - EDIÇÃO DE UMA LINHA DA TABELA CRIADA A PARTIR DE UM MODELO DO GEODJANGO. RETIRADO DO SITE DE ADMINISTRADOR DA PLATAFORMA DESENVOLVIDA.	23
FIGURA 11 - MAPA DO CAMPUS DA FCT UNL USANDO OPENSTREETMAP (ESQUERDA), POLÍGONOS DE NÍVEL 5 DA COS CORRESPONDENTES (DIREITA). RETIRADO DE [73]	25
FIGURA 12 - ALGORITMO PARA A LEITURA DOS DADOS DOS FICHEIROS EXCEL PARA <i>DATAFRAMES</i> DO PANDAS.....	30
FIGURA 13 - ALGORITMO PARA A CRIAÇÃO DA TABELA DE PERCENTAGEM DE ÁREA ARDIDA EM RELAÇÃO Á ÁREA DO PAÍS DE REGIÕES POR MEGA CLASSES DO ANO DE 2014.....	30
FIGURA 14 - ALGORITMO PARA A VERIFICAÇÃO DE DUPLICADOS NOS IDENTIFICADORES DOS POLÍGONOS DA COS....	31
FIGURA 15 - MODELO DO DJANGO USADO PARA A INSERÇÃO DOS DADOS DA TABELA 4	36
FIGURA 16 - CÓDIGO DA <i>VIEW</i> DA PÁGINA DOS GRÁFICOS DE MEGA CLASSE/ANO	37
FIGURA 17 - PÁGINA DOS GRÁFICOS DE MEGA CLASSE/ANO, MOSTRANDO O FORMULÁRIO	38
FIGURA 18 - CÓDIGO HTML DA PÁGINA DOS GRÁFICOS DE MEGA CLASSE/ANO.....	38
FIGURA 19 - UTILIZAÇÃO DA FERRAMENTA OGRINSPECT PARA GERAR O MODELO ONDE SERÁ GUARDADA A INFORMAÇÃO DO MAPA DE 2018	40
FIGURA 20 - CÓDIGO DA <i>VIEW</i> DE SERIALIZAÇÃO (FOGOS_2014_DATA) E DA <i>VIEW</i> DE REDIRECIONAMENTO (FOGOS_2014) DO ANO DE 2014.....	43
FIGURA 21 - CÓDIGO HTML DA APRESENTAÇÃO DO MAPA DE 2014.....	43
FIGURA 22 - EXEMPLO DO SERVIÇO DE VISUALIZAÇÃO DOS MAPAS, APRESENTANDO OS DADOS DO ANO DE 2017.....	44
FIGURA 23 - MENU DO SERVIÇO DE ANÁLISE DA INFORMAÇÃO DOS INCÊNDIOS	45
FIGURA 24 - FORMULÁRIO PARA ESCOLHA DO ANO/ANOS PARA OS QUAIS SE PRETENDA VISUALIZAR A INFORMAÇÃO	46
FIGURA 25 - FORMULÁRIO PARA ESCOLHA DA MEGA CLASSE(S) PARA AS QUAIS SE PRETENDA VISUALIZAR A INFORMAÇÃO	47
FIGURA 26 - FORMULÁRIO PARA ESCOLHA DA CLASSE(S) PARA AS QUAIS SE PRETENDA VISUALIZAR A INFORMAÇÃO	48
FIGURA 27 - FORMULÁRIO PARA ESCOLHA DAS REGIÕES PARA AS QUAIS SE PRETENDA VISUALIZAR A INFORMAÇÃO	49
FIGURA 28 - FORMULÁRIO PARA A ESCOLHA DO TIPO DE PERCENTAGEM QUE SE DESEJA VISUALIZAR	49

FIGURA 29 - GRÁFICO DA PÁGINA DE MEGA CLASSE/REGIÃO CONTENTO TODAS AS OPÇÕES SELECIONADAS	50
FIGURA 30 - GRÁFICO DA PÁGINA DE CLASSE/REGIÃO CONTENTO TODAS AS OPÇÕES SELECIONADAS.....	51
FIGURA 31 - GRÁFICO DA PÁGINA DE REGIÕES/ANO CONTENTO TODAS AS OPÇÕES SELECIONADAS	51
FIGURA 32 - GRÁFICO DA PÁGINA DE MEGA CLASSE/ANO CONTENTO TODAS AS OPÇÕES SELECIONADAS.....	52
FIGURA 33 - GRÁFICO DA PÁGINA DE CLASSE/ANO CONTENTO TODAS AS OPÇÕES SELECIONADAS	52
FIGURA 34 - MAPA COM A LOCALIZAÇÃO DAS ÁREAS AFETADAS PELOS INCÊNDIOS DE 2014	53
FIGURA 35 - MAPA COM A LOCALIZAÇÃO DAS ÁREAS AFETADAS PELOS INCÊNDIOS DE 2015	54
FIGURA 36 - MAPA COM A LOCALIZAÇÃO DAS ÁREAS AFETADAS PELOS INCÊNDIOS DE 2016	55
FIGURA 37 - MAPA COM A LOCALIZAÇÃO DAS ÁREAS AFETADAS PELOS INCÊNDIOS DE 2017	55
FIGURA 38 - MAPA COM A LOCALIZAÇÃO DAS ÁREAS AFETADAS PELOS INCÊNDIOS DE 2018	56

Lista de Tabelas

TABELA 1 - EXEMPLO DA INFORMAÇÃO DISPONÍVEL NAS 6 PRIMEIRAS ENTRADAS DO FICHEIRO DA COS2015	28
TABELA 2 - EXEMPLO DE ALGUMA DA INFORMAÇÃO DISPONÍVEL DAS 6 PRIMEIRAS ENTRADAS DO FICHEIRO DA INTERSEÇÃO DA COS2015 COM OS MAPAS DO ICNF DO ANO DE 2018.....	29
TABELA 3 - LISTA DOS POLÍGONOS COM A COMBINAÇÃO DE ID COM REGIÃO (<i>LAYER</i>) IGUAIS DOS DADOS DAS INTERSEÇÕES.....	32
TABELA 4 - PERCENTAGENS DE ÁREA ARDIDA EM RELAÇÃO Á ÁREA AFETADA DE MEGA CLASSE POR ANO.....	33

Lista de acrónimos e abreviações

- AP – Administração Pública
- API – *Application programming interface*
- COS – Carta de Uso e Ocupação do Solo
- CWI – *Centrum Wiskunde & Informatica*
- DGT – Direção Geral de Território
- DSF – *Django Software Foundation*
- ECMWF – *European Center for Medium-Range Weather Forecasts*
- EDO – *European Drought Observatory*
- EFAS – *European Flood Awareness System*
- EFFIS – *European Forest Fire Information System*
- ESA – Agência Espacial Europeia
- EUMETSAT – *European Organization for the Exploitation of Meteorological Satellites*
- FIRMS – *Fire Information for Resource Management System*
- ICNF – Instituto da Conservação da Natureza e das Florestas
- IPMA – Instituto Português do Mar e da Atmosfera
- MODIS – *Moderate Resolution Imaging Spectroradiometer*
- MTV – *Model-template-view*
- NUTS – *Nomenclature of Territorial Units for Statistics*
- OT – Observação da Terra
- SIG – Sistema de Informação Geográfica
- SQL – *Structures Query Language*
- UE - União Europeia
- VIIRS – *Visible Infrared Imaging Radiometer Suite*

1

Introdução

1.1. Contextualização

O Copernicus é um programa da União Europeia (UE) de Observação da Terra (OT). Este programa é liderado pela Agência Espacial Europeia (ESA) e tem à sua disposição uma família de 6 satélites de OT denominados de Sentinel [1]. A Direção-Geral do Território (DGT) juntamente com o Instituto Português do Mar e da Atmosfera (IPMA) desenvolveram uma infraestrutura chamada IPSentinel, que tem como objetivo armazenar e fornecer imagens dos satélites Sentinel do Copernicus, imagens estas que abrangem todo o território português [2]. No entanto, o volume crescente de dados provenientes da infraestrutura IPSentinel requer uma exigência acrescida para as agências da Administração Pública (AP), o que levou o IPSentinel a não ser amplamente utilizado pela AP. Devido a esta situação surgiu a necessidade de serem desenvolvidos mapas específicos para as agências da AP [2]. Alguns desses mapas são considerados produtos de valor acrescido, frequentemente chamados de produtos de nível 3 e têm um papel fundamental na gestão da ocupação do solo e no cumprimento dos compromissos internacionais do país [2].

A obtenção dos mapas é efetuada principalmente pela interpretação visual de imagens de satélite de alta resolução (Sentinel), o que exige uma alocação significativa de recursos humanos da AP, fazendo assim com que estes mapas levem muito tempo a serem produzidos. Por esse motivo os mapas sofrem de uma baixa taxa de atualização [2]. Devido a esta baixa taxa de atualização foi proposto o projeto IPSTERS pela DGT em parceria com o UNINOVA e com a Nova

Information Management Center (Magic center), tendo como objetivo principal a exploração das possíveis aplicações e limitações do uso de algoritmos de inteligência artificial para realizar o tratamento da informação proveniente da infraestrutura IPSentinel para a produção automática de produtos de nível 3, como os mapas da COS (Carta de uso e Ocupação do solo), reduzindo assim a necessidade de intervenção humana [2].

No caso da COS utilizada neste projeto, sendo estes mapas especificamente produzidos por interpretação manual de imagens tiradas por aviões de baixa altitude, os mapas são produzidos de 5 em 5 anos com uma resolução de 1 hectare. Os mapas CORINE, da UE, são produzidos pelo menos de 6 em 6 anos com uma resolução de 25 hectares [3, 4]. No entanto este tipo de mapas não são facilmente observados pelo público devido á necessidade da utilização de softwares específicos que os consigam ler. Deste problema surgiu a necessidade de existirem serviços na internet que permitam a visualização deste tipo de mapas. Em Portugal o Sistema Nacional de Informação Geográfica (SNGI) é a infraestrutura que permite o registo e a pesquisa de dados e serviços de dados geográficos [5]. Estes serviços portugueses, no entanto, não oferecem quaisquer conexões com as informações dos incêndios em Portugal, nem estatísticas sobre os mesmos. Surge portanto a necessidade de se criar plataformas que permitam o acesso a esse tipo de informações em junção com os mapas da COS.

1.2. Objetivos

Este trabalho teve como objetivo o desenvolvimento de uma plataforma *web* que permita ao público ter uma forma acessível de visualizar tanto dados estatísticos sobre os incêndios que decorreram no país desde 2014 até 2018, como vários mapas do território português onde se pode observar uma camada sobreposta no mapa com as áreas afetadas pelos incêndios em cada um desses mesmos anos. É de salientar que a análise quantitativa dos dados estatísticos não faz parte dos objetivos, ficando essa análise como possibilidade para um trabalho futuro. Os dados utilizados para a obtenção tanto da estatística com dos mapas foram obtidos usando a COS e os dados dos incêndios nacionais retirados do site do ICNF. Com a intenção de se ter conseguido alcançar o objetivo principal, este trabalho foi dividido em três partes: o tratamento dos dados, o desenvolvimento do serviço responsável por permitir a visualização dos dados estatísticos e o

desenvolvimento do serviço responsável pela visualização dos mapas dos incêndios.

No tratamento dos dados o objetivo foi o uso do *Pandas* [6], uma biblioteca de análise de dados da linguagem de programação Python. Esta biblioteca foi utilizada para se obter ficheiros com os dados organizados pelas diferentes categorias.

No desenvolvimento do serviço para a visualização dos dados estatísticos foi utilizada a *web Framework* Django. Esta foi escolhida para ser utilizada neste trabalho pela sua simplicidade e pela sua capacidade de utilizar várias aplicações, tais como o *django graphos* que permite a fácil implementação de diferentes tipos de gráficos.

No desenvolvimento do serviço para a visualização dos mapas dos incêndios foi utilizada a mesma *web Framework* tal como no serviço anterior, com a diferença no tipo de aplicações usadas, sendo que neste serviço o objetivo foi a utilização do *GeoDjango* e do *Django-leaflet* que permitiu a visualização de mapas interativos *leaflet* na aplicação.

Os objetivos deste trabalho encontram-se enquadrados no projeto IPSTERS mais especificamente na tarefa 6 do projeto que diz respeito á disseminação da tecnologia.

2

Estado de Arte

2.1. Fire Information for Resource Management System (FIRMS)

FIRMS é um sistema da NASA que disponibiliza informação sobre incêndios quase em tempo real [7]. Esta informação é disponibilizada aos utilizadores através de um mapa interativo do mundo que permite ao utilizador visualizar tanto dos incêndios ativos como de incêndios que já tenham ocorrido em anos anteriores. Este sistema funciona através da deteção de fogos ativos [7]. Esta deteção é realizada pela combinação entre o MODIS (*Moderate Resolution Imaging Spectroradiometer*) e o VIIRS (*Visible Infrared Imaging Radiometer Suite*) que disponibilizam para o FIRMS os dados dos incêndios num prazo de 3 horas após a passagem do satélite pela zona afetada e uma imagem do incêndio num prazo de 4-5 horas [7].

O MODIS é um instrumento chave que se encontra a bordo dos satélites Terra e Aqua. Este instrumento utiliza uma variedade de sensores para providenciar imagens da superfície do planeta, devido ao facto do sistema estar implementado nos dois satélites (Terra e Aqua). É assim possível para o MODIS obter imagens de toda a superfície do planeta a cada 1 a 2 dias, adquirindo dados em 36 bandas espectrais [8, 9].

O VIIRS é um instrumento da NASA para deteção de fogos ativos que foi lançado a 28 de outubro de 2011 a bordo do *Joint Polar Satellite System*. Este

instrumento detetou pela primeira vez um fogo ativo a 19 de janeiro de 2012 [7]. Os formatos dos dados dos fogos provenientes do VIIRS foram intencionalmente projetados com base no MODIS para suportar uma correspondência dos dados entre os dois instrumentos [10].

O FIRMS oferece ainda ao utilizador a opção de apresentar nos mapas os polígonos das áreas ardidas de um ano em específico ou ainda a opção de escolher o ano e o mês ou meses a apresentar, na Figura 1 podemos observar o exemplo dos polígonos das áreas ardidas de Portugal no ano de 2017 observado através da plataforma FIRMS.

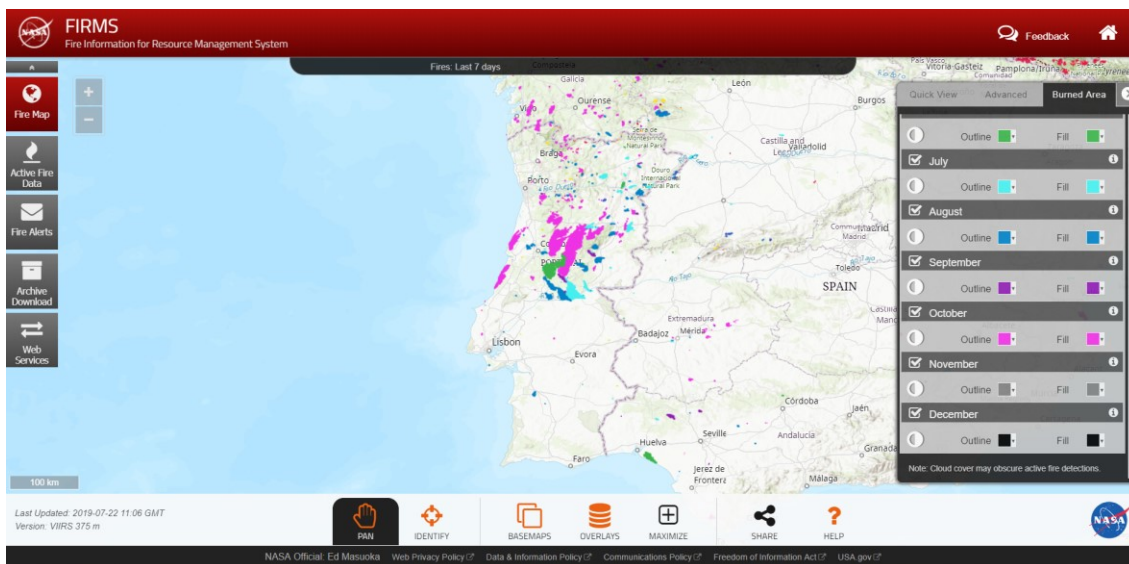


Figura 1 - Mapa das áreas ardidas de 2017 (FIRMS) [71]

2.2. Copernicus

Copernicus é o programa da União Europeia (UE) que tem como objetivo observar o planeta e o seu ambiente para o benefício dos cidadãos europeus [1]. O programa é coordenado e gerido pela comissão europeia e é implementado por uma parceria entre os estados membros da UE. Fazem parte deste consórcio, a agência espacial europeia (ESA), a organização europeia para a exploração de satélites Meteorológicos (EUMETSAT), o centro europeu de previsões meteorológicas de médio prazo (ECMWF) e o *Mercator Ocean* para além de outras agências europeias [1].

O programa Copernicus foi projetado especificamente com requisitos dos seus utilizadores em mente. Baseado em observações de satélites e no terreno, os serviços do Copernicus oferecem informação praticamente em tempo real numa escala global, com intuito de ajudar a melhor compreender o nosso planeta e ajudar a gerir o ambiente de forma sustentável [11].

Copernicus tem à sua disposição um conjunto de satélites dedicados. Este conjunto de satélites é conhecido como *the sentinel families*, que inclui o Copernicus Sentinel-1, Sentinel-2 (Figura 2), Sentinel-3, Sentinel-4, Sentinel-5 e Sentinel-5P. Este conjunto de satélites foi desenvolvido especificamente para satisfazer os requisitos dos serviços do Copernicus [11]. Os serviços do Copernicus processam e analisam os dados proveniente tanto dos satélites Sentinel como dos sensores presentes no terreno de modo a que os mesmos sejam usados na obtenção de conjunto de dados que contêm a informação já processada para que esta seja mais facilmente comparável e pesquisável [11]. Esta informação é obtida através dos seis seguintes serviços temáticos:

- Atmosfera;
- Marinha;
- Terra;
- Alterações Climáticas;
- Segurança;
- Emergência;



Figura 2 - Imagem do satélite Sentinel 2. Retirado de [72]

2.2.1. Serviço de gestão de emergências do Copernicus

O serviço de gestão de emergências do Copernicus ou Copernicus EMS fornece a todas as entidades envolvidas na gestão dos desastres naturais, crises humanitárias e desastres provocados pelo homem informações geoespaciais oportunas e precisas provenientes de sensores dos satélites Sentinel e completadas por fontes de dados abertas [12]. Este serviço inclui duas componentes:

- Uma componente de mapeamento;
- Uma componente de aviso antecipado;

A componente de mapeamento dispõe de uma cobertura global que fornece, às entidades do projeto Copernicus, mapas baseados em imagens de satélites. Estes mapas podem ser visualizados em formato digital ou físico, dispondo também da capacidade de serem combinados com outras fontes de dados para suportarem uma análise geoespacial mais completa [12].

A componente de aviso antecipado consiste em três sistemas diferentes [12]:

- *European Flood Awareness System* (EFAS)
- *European Forest Fire Information System* (EFFIS)
- *European Drought Observatory* (EDO)

O EFAS é o sistema que fornece perspectivas gerais sobre inundações que estejam atualmente a decorrer e fornece também previsões sobre futuras inundações com até 10 dias de antecedência. O EFFIS é o sistema que permite que forneça informação histórica ou praticamente em tempo real dos incêndios florestais nas regiões da Europa, do Médio Oriente e do Norte de Africa. O EDO é o sistema que fornece informações relevantes sobre as secas a nível Europeu.

Além destes três serviços da componente de aviso antecipado do Copernicus EMS, existem ainda três outros serviços que completam a cobertura dos serviços do Copernicus a um nível global, sendo estes o *Global Flood Awareness System*, *Global Wildfire Information System* e *Global Drought Observatory*.

2.2.1.1. *European Forest Fire Information System* (EFFIS)

O EFFIS é o sistema da união europeia (UE) que presta apoio aos serviços encarregues da proteção dos fogos florestais nos países da UE. Este sistema tem também como objetivo fornecer informação atualizada e fidedigna sobre os fogos florestais ao parlamento europeu [13]. O EFFIS foi estabelecido pela comissão europeia e está a ser apoiado desde 1998 por uma rede de especialistas provenientes dos diversos países da UE. Esta rede de especialistas é chamada *Expert Group on Forest Fires*, e está registada na secretaria geral da união europeia [14]. Este grupo atualmente é formado por especialistas de 43 países da Europa, do Médio Oriente e do Norte de Africa [14].

EFFIS tem à sua disposição diversos módulos, cujos resultados são usados nas aplicações disponibilizadas pelo EFFIS. Estes módulos são [15]:

- *Fire Danger Forecast* – responsável pelos mapas com níveis de perigo previsto de incêndio usando previsões meteorológicas numéricas [16];
- *Active Fire Detection* – responsável pelo mapeamento da informação fornecida pela NASA FIRMS [17];
- *Rapid Damage Assesment* – responsável pelo mapeamento das áreas ardidas usando uma análise diária das imagens fornecidas pelo MODIS [18];
- *Fire Damage Assesment* – responsável pelo mapeamento das áreas ardidas no final de cada época de fogos [19];
- *European Fire Database* – contém a base de dados com informação sobre os fogos florestais compilada pelos estados membros da UE [20];
- *Seasonal Forecast* – pretende realizar uma previsão sobre o clima para um período de até 7 meses [21];
- *Monthly Forecast* – pretende realizar uma previsão sobre o clima para um período de até 15 dias [22];
- *Fuels* – responsável pelo mapa de combustíveis da Europa [23];

O EFFIS oferece aos seus utilizadores uma variada gama de aplicações, tais como [24]:

- *Current Situation*;

- *Long-term monthly fire weather forecast;*
- *Long-term seasonal fire weather forecast;*
- *Fire News;*
- *Data and services;*

Current Situation é uma das aplicações do EFFIS mais usadas [13]. Esta aplicação oferece ao utilizador um mapa interativo onde este pode observar os dados dos vários módulos que estão incorporados no EFFIS, como o *Fire Danger Forecast*, o *Rapid Damage Assessment* e as ferramentas de análise. O módulo do *Fire Danger Forecast* está presente na aplicação na forma de uma layer adicional que apresenta várias classificações de perigo que vão de muito baixo a extremo. Podemos observar um exemplo deste módulo na Figura 3.

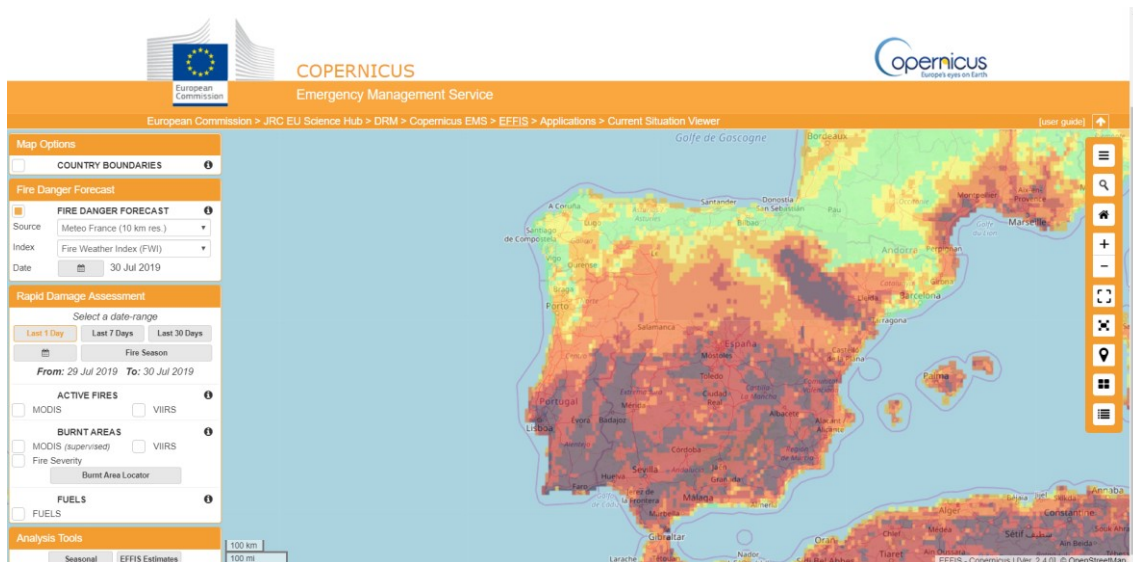
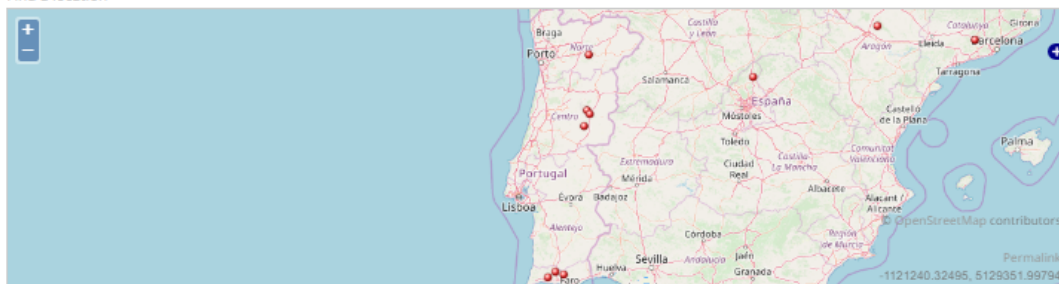


Figura 3 – Interface do EFFIS – aplicação do Current Situation - Fire Danger. Retirado de Forecast [76]

Fire News é uma das aplicações do EFFIS mais usadas [13], esta aplicação disponibiliza ao utilizador uma seleção de notícias sobre fogos florestais e um mapa interativo onde se podem observar as localizações dos fogos mencionados nas notícias. Na Figura 4 podemos observar a interface desta aplicação.

Fire News

Find a location



Search: From: To:

Country	Place	Size	Update	News
Algeria	Bainem, Algiers		29/07/2019	1 News linked
Algeria	Boghni	30	26/07/2019	1 News linked
Algeria	Guentoufa, Taret		29/07/2019	1 News linked
Algeria	Jijel	54	29/07/2019	1 News linked
Algeria	Saharidj	100	29/07/2019	1 News linked
Algeria	Tifra		24/07/2019	1 News linked
Croatia	Dubrava, Šibenik	900	29/07/2019	4 News linked
Croatia	Zadar	10	25/07/2019	1 News linked
Cyprus	Ayia Napa	10	29/07/2019	1 News linked
Cyprus	Pyrgos/Parekkklisia	7	24/07/2019	1 News linked

90 fires selected (with 103 related news). Get this KML

Size Class: **All** - N.A. - Small - Medium - Large - Major

Country: **All** - France - Greece - Italy - Portugal - Spain - Turkey -

Figura 4 – Interface do EFFIS – aplicação do *Fire News*. Retirado de [75]

2.3. Fogos.pt

Fogos.pt é um *website* construído pelo programador João Pina (Tomahock), que disponibiliza ao público de uma forma acessível a visualização do estado tanto dos fogos que estejam a decorrer atualmente no país, como dos fogos que tenham concluído nas últimas 24 horas. Esta informação é retirada do portal online da proteção civil, informação esta que é atualizada a cada 2 minutos [25]. O *website* disponibiliza esta informação na forma de um mapa interativo onde é possível ao utilizador selecionar um incêndio para ter acesso aos seus detalhes. Estes detalhes incluem a localização, a natureza, as datas em que os fogos se tenham iniciado, do início da atuação dos bombeiros e da conclusão. Além destas datas o utilizador pode ainda observar o número de meios (humanos, terrestres e

aéreos) enviados para o local. Um exemplo desta funcionalidade do *website* pode ser observado na Figura 5.

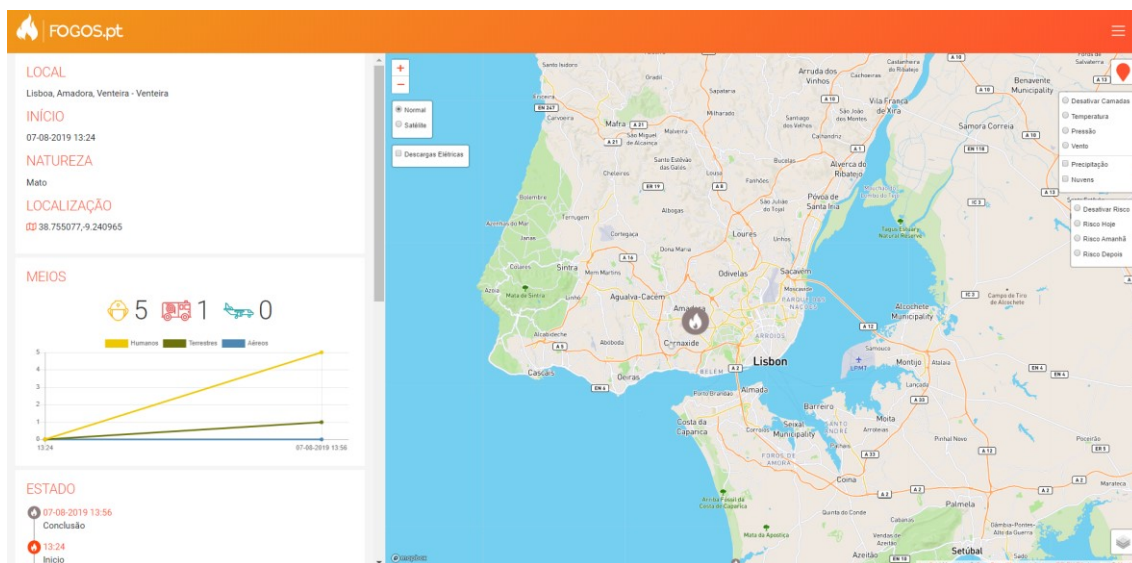


Figura 5 – Interface do *website* Fogos.pt - Detalhes de um incêndio. Retirado de [73]

Além das informações disponíveis a partir do mapa, o Fogos.pt ainda disponibiliza aos seus utilizadores uma secção de estatística onde se pode observar o número total de meios ativos no país, o número de incêndios que ocorreram tanto no dia atual como no dia anterior, contendo também o número de incêndios por região. Na Figura 6 pode ser observada a estatística visualizada no dia 07 de Agosto de 2019.



Figura 6 – Interface do *website* Fogos.pt – Detalhes da estatística do dia 07 de agosto de 2019. Retirado de [77]

3

Metodologia

Neste capítulo serão apresentadas as várias ferramentas e aplicações que tornaram este projeto possível, tais como a linguagem de programação Python, a *web framework* Django, a COS e os mapas do ICNF.

3.1. Python

Python é uma linguagem de programação de alto nível e de uso geral, disponibilizada pela primeira vez em 1991 por Guido van Rossum [26]. Esta linguagem de programação enfatiza a legibilidade do código com o uso de indentação [26]. A abordagem do Python à programação é uma abordagem orientada a objetos, com o objetivo de ajudar os programadores a escreverem código claro e lógico tanto para projetos de pequena escala como para projetos de grande escala [26].

A linguagem Python é uma linguagem dinâmica, pelo facto de que quando esta está em execução, efetua muitos procedimentos que outras linguagens de programação estática executam apenas durante a compilação do código [27, 28].

3.1.1. Breve história do Python

A linguagem Python foi criada por Guido van Rossum no Centrum Wiskunde & Informatica (CWI) na Holanda como sucessora da linguagem ABC, com a capacidade de processar exceções e trabalhar com o sistema operacional Amoeba [29, 30]. Desde o início da implementação do Python em dezembro de 1989 até 12 de julho de 2018, Guido van Rossum foi sempre quem esteve a liderar o desenvolvimento do Python. Foi devido a todos estes anos de comprometimento da parte do Van Rossum que este recebeu o título de “*Python's Benevolent Dictator For Life*” pela comunidade Python [31, 32].

Ao longo dos anos a linguagem Python foi recebendo várias atualizações. Entre estas atualizações, duas delas podem ser consideradas as mais significativas:

- Python 2.0, lançado a 16 de outubro de 2000 [33];
- Python 3.0, lançado a 3 de dezembro de 2008 [34];

3.2. Django

O Django é um *web framework* grátis que disponibiliza o seu código no formato de *open-source*, este *framework* é baseada em python, e segue o modelo de arquitetura de MTV ou *model-template-view* [35, 36]. O desenvolvimento do Django está a cargo da Django Software Foundation (DSF), uma organização independente, sem fins lucrativos [37].

O principal objetivo do Django é proporcionar uma forma fácil de criar sites complexos. Para alcançar este objetivo o Django está desenvolvido para dar ênfase à capacidade reutilização de componentes para permitir uma programação com pouco código e com um desenvolvimento rápido. Além disso o Django segue ainda os princípios de consistência e de que explícito é melhor que implícito [38]. O princípio de consistência no Django é observado a todos os níveis de desenvolvimento desde o baixo nível (código em python) até ao alto nível (a “experiência” de usar o Django), enquanto o princípio de que explícito é melhor que implícito vem do python, sendo este um princípio central do python. No Django este princípio pode ser observado de modo em que o Django não faça “magia”

pelo programador de forma em que tudo o que o Django faz está de uma forma ou outra explícita para o programador [38].

Em seguida será apresentada uma breve história de como o Django surgiu e a filosofia dos seus criadores no que respeita às várias partes responsáveis pelo desenvolvimento do site. Serão ainda explicadas as várias aplicações que foram utilizadas em junção com o Django para a criação do site desenvolvido neste projeto. Na Figura 7 pode ser observada a primeira página que é apresentada ao utilizador após a criação de um novo projeto Django. Esta figura foi captada pelo autor durante a fase inicial do projeto.

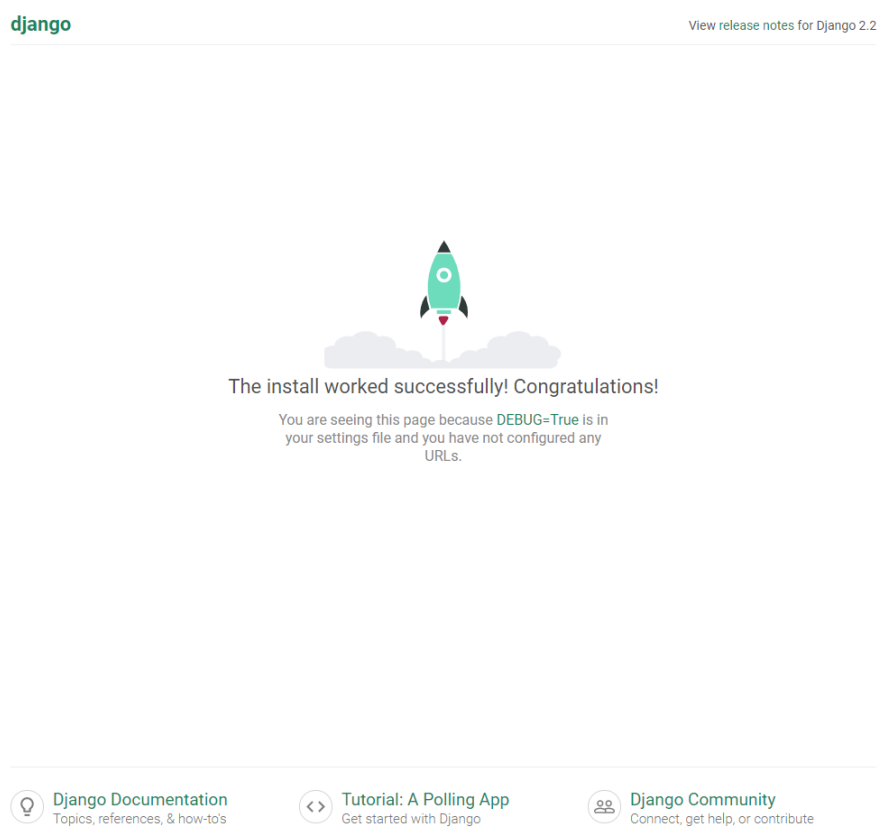


Figura 7 - Página inicial do Django,

3.2.1

3.2.1. Breve história do Django

O Django surgiu de uma necessidade do World Online, uma operação de uma plataforma *web* de jornalismo, responsável pela criação de aplicações *web* com pequenos prazos, em que apenas numa questão de horas era preciso levar um conceito de uma aplicação até ao seu lançamento para o público [35].

Foi no Outono de 2003 que dois programadores do World Online (Adrian Holovaty and Simon Willison) deixaram de usar PHP e começaram a usar Python para desenvolver sites na internet [35]. Ao criar sites altamente interativos e intensivos, como o Lawrence.com, eles começaram a extrair dos sites uma estrutura genérica que lhes permitia criar aplicações *web* cada vez mais rápido [35]. Ao longo de um período de dois anos esta estrutura foi constantemente melhorada [35].

No Verão de 2005, o World Online decidiu tornar o software livre, disponibilizando o código gratuitamente na internet. Este software começou então a chamar-se Django, estando o nome relacionado com Django Reinhardt, um guitarrista de *Gypsy jazz* [35].

Atualmente existem muito sites que são desenvolvidos com a *framework* Django. Entre os inúmeros sites destacam-se alguns nomes como o Instagram, Disqus, Mozilla e NASA [39].

3.2.2 Filosofias do Django

Durante o desenvolvimento do Django os seus criadores seguiram várias filosofias fundamentais que foram aplicadas nos vários aspetos da *framework*, entre os aspetos afetados podem encontrar-se os seguintes [38]:

- Modelos;
- API da base de dados;
- *Design* de URL;
- Sistema de modelos;
- *Views*;
- Estrutura de *cache*;

Os modelos no Django são usados para descrever as tabelas da base de dados usando apenas Python. Neste aspecto o Django adota o princípio do Python em que algo explícito é melhor que algo implícito. Este princípio é usado nos vários campos dos modelos, no sentido em que estes não assumem comportamentos baseados apenas nos nomes dos campos porque assim podiam levar a uma variedade de erros, necessitando também de um conjunto de argumentos juntos com o tipo de campo [38].

A interface de programação de aplicações (API) da base de dados no Django foi desenvolvida com três filosofias em consideração, eficiência do SQL, eficácia da sintaxe e a opção de usar SQL puro quando necessário [38]. Estas filosofias podem ser observadas no Django, quando este efetua comandos de consulta para a base de dados, comandos esses que podem ser feitos pelo utilizador usando apenas Python com uma sintaxe concisa, mas eficaz [38].

O *design* de URL no Django foi desenvolvido com os utilizadores em consideração, oferecendo uma flexibilidade infinita na escolha dos nomes dos URL. Esta possibilidade vem do facto de que o Django não associa o nome do URL a uma função permitindo assim a sua flexibilidade [38].

O sistema de modelos ou *template system* do Django foi concebido para facilitar o trabalho do programador no desenvolvimento da apresentação do site, para alcançar este objetivo o sistema de modelos controla apenas a lógica de apresentação do site, lógica esta que oferece ao programador a capacidade de fazer com que uns modelos de apresentação herdem outros modelos eliminando assim a duplicação do código de apresentação. Além desta funcionalidade o sistema de modelos permite ainda que os modelos não sejam gerados apenas com HTML, podendo ser usados outros formatos, como o Django *Template Language* (DTL) que é uma linguagem de programação exclusiva do Django, que usa uma sintaxe simples reconhecida pelo sistema de modelos [40, 38].

As *Views* no Django são um aspeto da *framework* que trabalha com as solicitações tanto da forma GET como POST. Estas foram desenvolvidas com a sua simplicidade em mente, sendo estas escritas apenas como uma simples função de Python, que funciona independentemente do modelo de apresentação que lhes está associado [38].

No Django a estrutura de *cache* tem três princípios fundamentais, sendo estes [38]:

- Menos código – a *cache* deve funcionar o mais rápido possível, levando a que o código de *backend* da mesma seja mantido no mínimo possível;
- Consistência – a API da *cache* deve proporcionar uma interface consistente nos diferentes *backends* da mesma;
- Extensibilidade – a API da *cache* deve ser extensível ao nível da aplicação com base nas necessidades do programador;

3.2.3 Graphos

Graphos é uma aplicação do Django que normaliza dados para criar gráficos [41]. Esta aplicação oferece uma maneira simples e concisa de usar JavaScript para trabalhar com diversos tipos de gráficos, permitindo ao programador mudar completamente o tipo de gráfico em poucos passos (ver na Figura 8 exemplos de gráficos criados por esta aplicação) [41].

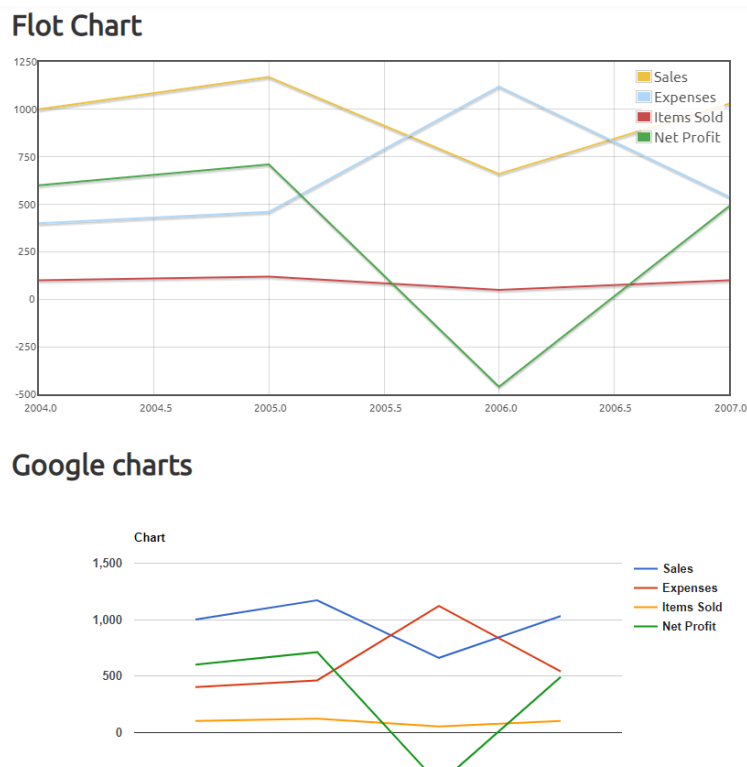


Figura 8 - Exemplos de gráficos gerados pelo Graphos

3.2.4 Django Leaflet

Django Leaflet é um pacote de Python para a biblioteca de Javascript do Leaflet, que permite a integração de mapas interativos do Leaflet com o Django,

sem necessitar de incorporar todos os recursos do Leaflet em todos os projetos Django [42]. Este pacote permite ainda a customização dos mapas a partir das definições do Django [42].

A biblioteca Leaflet de onde este pacote de Python é retirado, é uma biblioteca de JavaScript para mapas interativos compatíveis com dispositivos móveis. Esta biblioteca foi desenhada tendo a consideração a simplicidade, o desempenho e a facilidade de utilização [43].



Figura 9 - Vista do mapa do campus da FCT-UNL usando Leaflet

3.2.5 GeoDjango

O GeoDjango é um módulo que está incorporado no Django e que lhe permite tornar-se uma *framework web* geográfica de escala mundial. Este módulo tem como objetivo principal facilitar ao máximo a criação de aplicações SIG (Sistema de Informação Geográfica) [44].

Este módulo do Django tem uma desvantagem importante pois não se consegue utilizar sem antes se ter instalado diversas bibliotecas e programas extra. Estes requisitos variam em função da base de dados escolhida para guardar os dados geográficos. No entanto esta desvantagem não afeta um utilizador

normal do site, afetando apenas o programador [45]. No caso da base de dados PostgreSQL, o GeoDjango necessitou dos seguintes requisitos [45]:

- GEOS – uma biblioteca em C++ para executar operações geométricas [46];
- GDAL – uma biblioteca de funcionalidades de dados SIG, entre as inúmeras funcionalidades destacam-se a capacidade de ler e gravar dados geográficos vetoriais em diversos formatos [47];
- PROJ.4 – um programa de transformação de coordenadas geoespaciais de um sistema de referência de coordenadas (SRC) para outro [48];
- PostGIS – uma extensão da base de dados PostgreSQL para que esta obtenha a capacidade de guardar objetos geográficos permitindo o uso de consultas á base de dados deste tipo de dados em SQL [49];

O GeoDjango após ter todos os seus requisitos instalados oferece ao programador uma variedade de ferramentas e API, entre elas podem se encontrar as seguintes:

- GeoDjango *Model* API;
- GeoDjango *Database* API;
- GeoDjango *Forms* API;
- GEOS API;
- GDAL API;
- Ferramentas GeoDjango;

A GeoDjango *Model* API permite ao programador construir modelos no Django com vários tipos de campos espaciais [50]. Estes campos espaciais consistem numa série de campos de geometria e um *raster field* [50]:

- *GeometryField* – a classe base para campos de geometria;
- *PointField* – guarda um ponto;
- *LineStringField* – guarda uma sequência de pontos;
- *PolygonField* – guarda um polígono;
- *MultiPointField* – guarda um conjunto de pontos;
- *MultiLineStringField* – guarda um conjunto de sequências de pontos;
- *MultiPolygonField* – guarda um conjunto de polígonos;
- *GeometryCollectionField* – guarda um conjunto que pode ser composto por vários tipos de campos de geometria diferentes;

- *RasterField* – guarda um construtor GDALRaster que tem a função de criar dados *raster*, um tipo de dados que consiste numa matrix de células onde cada célula contém um valor que representa informação, como temperatura [47, 51];

A GeoDjango *Database* API é responsável pela transformação do código em Python escrito no Django para código SQL reconhecido pelas várias bases de dados suportadas pelo GeoDjango [52]. Estas funcionalidades incluem a transformação dos modelos do Django para tabelas da base de dados e das pesquisas feitas em Python no Django para SQL *queries* [52].

A GeoDjango *Forms* API providencia a ferramenta de construção de formulários do Django com um conjunto de campos e *widgets* especializados para a visualização e edição de dados geográficos [53]. Ao conjunto de campos disponibilizados por esta ferramenta foram adicionados dois argumentos a cada um dos campos. O argumento SRID indica o código SRID para o qual devem ser transformadas as coordenadas introduzidas no campo associado e o argumento *geom_type* indica o tipo de geometria. Este último argumento geralmente não deve ser usado pelo facto de que a sua predefinição depende do tipo de campo a que está associado [53].

Os campos de geometria adicionados à ferramenta de formulários do Django são:

- *GeometryField*;
- *PointField*;
- *LineStringField*;
- *PolygonField*;
- *MultiPointField*;
- *MultiPolygonField*;
- *GeometryCollectionField*;

Os *widgets* que são disponibilizados com esta ferramenta permitem a edição e visualização dos dados geográficos num mapa. De momento estes *widgets* não suportam geometrias 3D. Os *widgets* disponíveis são os seguintes:

- *BaseGeometryWidget* – este *widget* contém a lógica necessária pelas subclasses, no entanto não pode ser usado diretamente para campos de geometria;
- *OpenLayersWidget* – é o *widget* padrão usado por todos os campos de geometria do GeoDjango;
- *OSMWidget* – este *widget* utiliza a camada base *OpenStreetMap* para a visualização de objetos geométricos;

A GEOS API é um *wrapper* Python de alto nível implementado pelo GeoDjango da biblioteca GEOS [54]. GEOS significa *Geometry Engine – Open Source* e é uma biblioteca C++, que implementa recursos simples do *OpenGIS* para SQL de funções de atributos e operadores espaciais [54]. Esta API tem as subseqüentes características [54]:

- Interface licenciada pela BSD para as rotinas de geometria GEOS;
- Interface que pode ser utilizada fora do Django;
- Mutabilidade, por permitir alterar objetos de *GEOSGeometry*;
- Compatível com os vários sistemas operativos;

A GDAL API do Django é uma interface Python de alto nível. No entanto mesmo contendo o nome de GDAL esta interface contém algumas das capacidades de um subconjunto do GDAL chamado de *OGR Simple Features Library* e contém ainda um suporte mínimo para recursos do GDAL em relação a dados *raster* [47]. As capacidades disponibilizadas por esta interface do OGR são a leitura e transformação de coordenadas de dados espaciais vetoriais [47].

Além das várias API descritas em cima, o GeoDjango dispõe ainda de uma gama variada de ferramentas, tais como [55]:

- *LayerMapping* – fornece uma forma de mapear o conteúdo de ficheiros com dados espaciais vetoriais para os modelos GeoDjango [56];
- *Ogrinspect* – disponibiliza uma maneira de inspecionar ficheiros compatíveis com OGR (ex. *shapefiles*) [57];

- GeoDjango *admin site* – permite ao Django aceder aos modelos do GeoDjango, no site de administrador, e visualizar e editar os dados espaciais num mapa interativo (ver Figura 10) [58];
- *KMZSitemap* – permite apresentar dados geolocalizados em formatos legíveis por máquinas [59];

Django administration WELCOME, ADMIN / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / World / World borders / Portugal

Change world border HISTORY

Name:

Area:

Population 2005:

FIPS Code:

2 Digit ISO:

3 Digit ISO:

United Nations Code:

Region Code:

Sub-Region Code:

Lon:

Lat:

Mpoly:

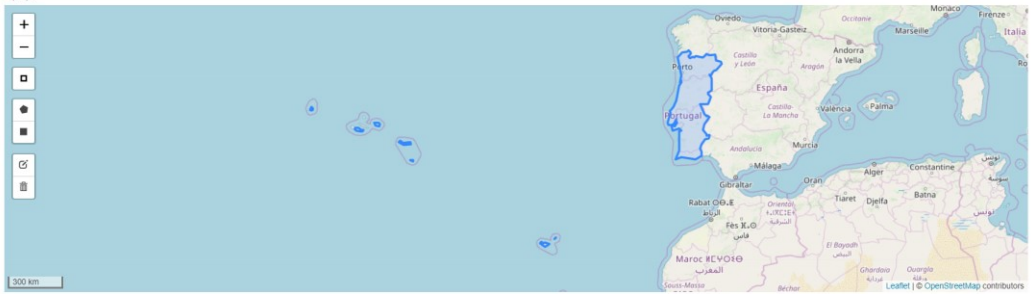


Figura 10 - Site de Administrador do Django - Edição de uma linha da tabela criada a partir de um modelo do GeoDjango. Retirado do site de administrador da plataforma desenvolvida.

3.3. COS

A carta de ocupação de solo (COS) é uma cartografia nacional da responsabilidade da Direção Geral do Território (DGT). Esta é uma cartografia

temática que tem como objetivo caracterizar com grande detalhe a ocupação/uso do solo no território de Portugal continental [60]. Atualmente a COS dispõe de quatro versões (1995, 2007, 2010 e 2015), sendo que de entre as versões disponíveis este projeto utilizou a versão mais recente (COS2015) para o seu desenvolvimento [61]. As cartografias de ocupação/uso do solo desempenham um papel fundamental no ordenamento do território e na monitorização ambiental. É por este motivo que se tem vindo a alargar a aplicação deste tipo de cartografias ao planeamento ambiental, político, económico e social [62].

A COS2015 é uma cartografia que contém a informação em formato vetorial. Esta informação é exposta em forma de polígonos. Para que um polígono possa ser representado na COS tem de obedecer aos seguintes requisitos:

- Possuir uma área superior a 1 hectare;
- Uma distância entre linhas igual ou superior a 20m;
- Uma percentagem de classe de uso de solo superior a 75%;
- Escala equivalente a 1:25 000;

Cada um dos polígonos da COS contém apenas uma classe. A COS2015 é uma cartografia constituída por um sistema hierárquico de classes de ocupação do solo, possuindo cinco níveis diferentes em complexidade, dos quais se destacam o 1º nível com 9 classes (agricultura, corpos de água, espaços descobertos ou com vegetação esparsa, florestas, matos, pastagens, sistemas agroflorestais, territórios artificializados, zonas húmidas) e o 5º nível com 48 classes [62]. A exatidão temática de cada um destes 5 níveis, com um erro de 2% para um nível de confiança de 95%, é de:

- 1º nível – 96%;
- 2º nível – 90%;
- 3º nível – 88%;
- 4º nível – 88%;
- 5º nível – 83%;



Figura 11 - Mapa do campus da FCT UNL usando OpenStreetMap (esquerda), polígonos de nível 5 da COS correspondentes (direita). Retirado de [74]

3.4.

Instituto da conservação da natureza e das florestas

O instituto da conservação da natureza e das florestas (ICNF) é um instituto público que se encontra integrado na administração indireta do estado. Sendo este um instituto dotado de autonomia administrativa, financeira e património próprio, esta instituição foi criada em 2012 pelo Ministério da Agricultura, do Mar, do Ambiente e do Ordenamento do Território em resultado da fusão da autoridade florestal nacional com o instituto da conservação da natureza e da biodiversidade [63, 64]. O ICNF tem como missão propor, acompanhar e assegurar a execução das políticas encarregues da conservação da natureza e das florestas, visando a utilização sustentável, a conservação, a fruição e o reconhecimento público do património natural [64]. Este instituto tem ainda como missão promover o desenvolvimento sustentável dos espaços florestais e dos recursos associados, assegurando a prevenção estrutural no quadro do planeamento e atuação no domínio da floresta e dos recursos cinegéticos e aquícolas das águas interiores e outros associados diretamente às florestas [64].

O ICNF disponibiliza os mapas dos incêndios rurais de diversos anos na plataforma (<http://www2.icnf.pt/portal/florestas/dfci/inc/mapas>). Estes mapas fazem parte da missão do ICNF de conservação da natureza e das florestas, mais especificamente da defesa da floresta contra incêndios. Estes mapas são disponibilizados ao público de duas formas, uma simples imagem que dá ao público a noção de onde ocorreram os incêndios sem grande detalhe e um conjunto de ficheiros do tipo *ShapeFile* (.shp) que permite a visualização mais detalhada das zonas onde ocorreram os incêndios. Esta segunda forma é menos

acessível ao público pois necessita que sejam utilizados programas preparados para ler esse tipo de ficheiros.

4

Desenvolvimento

Neste capítulo será apresentado o trabalho realizado durante o desenvolvimento deste projeto. Esta apresentação será dividida em 3 partes, começando pela apresentação da forma como foi realizado o tratamento dos dados, seguida do desenvolvimento dos serviços da plataforma web em Django, sendo estes o serviço para a análise da informação dos incêndios e o serviço responsável pela visualização dos mapas dos incêndios. A versão do Django usada neste projeto foi a versão 2.2.2 lançada a 3 de junho de 2019 [65].

4.1. Tratamento de dados

A primeira fase do desenvolvimento deste projeto começou pelo tratamento dos dados, que consistiu na análise de toda a informação disponibilizada e pela seleção da informação mais relevante para ser apresentada na plataforma web. A informação disponibilizada para a realização desta fase do projeto consistiu num conjunto de ficheiros CSV com toda a informação dos polígonos da COS2015 e com a informação da interseção dos polígonos da COS2015 com os polígonos provenientes dos mapas de incêndios do ICNF dos anos de 2014 a 2018.

No ficheiro que contém a informação dos polígonos da COS2015, do qual a Tabela 1 foi retirada, pode ser observada uma grande variedade de informação, tal como a classe de nível 1 (MEGACLASSE) e nível 5 (COS2015_LE) a que cada polígono se encontra atribuído, a área em hectares (ha) de cada polígono, o ID atribuído pela COS e o código da região a que esse polígono pertence (*layer*).

Este ficheiro contém esta informação para todos os 622629 polígonos da COS2015.

Os ficheiros com os dados da interseção dos polígonos dos mapas dos fogos dos anos de 2014 a 2018 com os polígonos da COS2015, de onde foram retirados os dados para a

Tabela 2, contém uma junção dos dados tanto dos polígonos da COS, como dos polígonos referentes aos incêndios e contendo ainda os dados da interseção. Estes dados estão apresentados nos ficheiros da forma em que cada linha da tabela contém toda a informação do polígono resultante da interseção. Esses dados incluem:

- As classes de nível 1 (Mega classes) e 5 (classes) da COS;
- Área do polígono da COS;
- Área do incêndio;
- Área da interseção, entre a área do polígono da COS com a área do incêndio;
- ID do polígono da COS;
- ID do polígono do incêndio;
- Localização do incêndio;
- Região da COS (*layer*).

Tabela 1 - Exemplo da informação disponível nas 6 primeiras entradas do ficheiro da COS2015

FID	COS2015_V1	COS2015_LE	MEGACLASSE	AREA	ID	LAYER
1	1.1.1.0 0.0	Tecido urbano contínuo	Territórios artificializados	1.009 379	116 95	COS2015-V1- PT111_Alto_Minho
2	1.1.1.0 0.0	Tecido urbano contínuo	Territórios artificializados	15.01 829	117 85	COS2015-V1- PT111_Alto_Minho
3	1.1.1.0 0.0	Tecido urbano contínuo	Territórios artificializados	1.595 837	118 13	COS2015-V1- PT111_Alto_Minho
4	1.1.1.0 0.0	Tecido urbano contínuo	Territórios artificializados	1.383 250	118 19	COS2015-V1- PT111_Alto_Minho
5	1.1.1.0 0.0	Tecido urbano contínuo	Territórios artificializados	2.809 189	118 22	COS2015-V1- PT111_Alto_Minho
6	1.1.1.0 0.0	Tecido urbano contínuo	Territórios artificializados	4.185 463	118 24	COS2015-V1- PT111_Alto_Minho
...

Tabela 2 - Exemplo de alguma da informação disponível das 6 primeiras entradas do ficheiro da interseção da COS2015 com os mapas do ICNF do ano de 2018

FID	COS2015_LE	MEGACLASS E	CALC_ AREA_ COS	COS_ID _FID	AREA_CALC_ INTERSECT	LAYER
1	Tecido urbano descontínuo	Territórios artificializad os	2.5705 2	2312	0.44047	COS2015-V1- PT111_Alto_ Minho
2	Tecido urbano descontínuo	Territórios artificializad os	1.3025 7	2337	1.30257	COS2015-V1- PT111_Alto_ Minho
3	Tecido urbano descontínuo	Territórios artificializad os	9.0102 0	2349	0.05847	COS2015-V1- PT111_Alto_ Minho
4	Tecido urbano descontínuo	Territórios artificializad os	1.2023 8	2365	1.20238	COS2015-V1- PT111_Alto_ Minho
5	Áreas de extracção de inertes	Territórios artificializad os	13.444 4	2775	11.6765	COS2015-V1- PT111_Alto_ Minho
6	Culturas temporárias de sequeiro e regadio	Agricultura	2.5814 0	3220	1.05403	COS2015-V1- PT111_Alto_ Minho
...

Devido ao grande número de dados disponíveis foi necessário recorrer ao desenvolvimento de vários algoritmos para efetuar a leitura dos ficheiros e a condensação dos dados para poderem ser introduzidos na base de dados da plataforma web desenvolvida. Estes algoritmos foram desenvolvidos recorrendo à linguagem Python e em muitos casos, à biblioteca Pandas [6]. Na Figura 12 pode-se observar o algoritmo utilizado na leitura dos ficheiros e na Figura 13 pode-se observar um dos algoritmos utilizados para transformar os dados em tabelas prontas a serem introduzidas na base de dados. Os restantes algoritmos são observados no Anexo I.

```

#
# Read and prepare data
#

#Required imports
from plotly.offline import iplot, init_notebook_mode
import pandas as pd
import pandas_profiling
import matplotlib
import numpy as np
import os
from math import sqrt

# set working folder
workingPath='D:/Universidade/TESE/'

# Read data from excel to DataFrames
ardidoCOS2014=pd.read_excel(workingPath+'Fogos_interseccao2014.xlsx') # Interception between COS and Burned2014
ardidoCOS2015=pd.read_excel(workingPath+'Fogos_interseccao2015.xlsx') # Interception between COS and Burned2015
ardidoCOS2017=pd.read_excel(workingPath+'Fogos_interseccao2017.xlsx') # Interception between COS and Burned2016
ardidoCOS2016=pd.read_excel(workingPath+'Fogos_interseccao2016.xlsx') # Interception between COS and Burned2017
ardidoCOS2018=pd.read_excel(workingPath+'Fogos_interseccao2018.xlsx') # Interception between COS and Burned2018

COS2015=pd.read_excel(workingPath+'cos_Portugal_2015.xlsx') # COS 2015

```

Figura 12 - Algoritmo para a leitura dos dados dos ficheiros Excel para DataFrames do Pandas

```

#####
#Create Data frame with Regions/Mega - Perc_country
#####

cos_codes=COS2015.layer.unique()
cos_mega=COS2015.Megaclasse.unique()
#####
#create aux data frame to eliminate duplicates
ardidos_area=ardidoCOS2014.copy()
ardidos_area.drop_duplicates(subset ="COS_ID_fid", keep = "first", inplace = True)
#####
Areas = pd.DataFrame(index=cos_codes)

for k in cos_mega:
    Areas[k]=0.0
for i in cos_codes:
    for j in cos_mega:
        total=ardidos_area[(ardidos_area.layer==i) & (ardidos_area.Megaclasse==j)].AREA.sum()
        total_country=COS2015[(COS2015.layer==i) & (COS2015.Megaclasse==j)].AREA.sum()
        burned=ardidoCOS2014[(ardidoCOS2014.layer==i) & (ardidoCOS2014.Megaclasse==j)].area_calc_intersect.sum()
        if (total!=0):
            Areas[j][i]=burned/total_country*100
        else:
            Areas[j][i]=0
Areas=Areas.sort_index()
Areas

```

Figura 13 - Algoritmo para a criação da tabela de percentagem de área ardida em relação á área do país de regiões por Mega classes do ano de 2014

Os algoritmos desenvolvidos para a criação das tabelas que seriam introduzidas na base de dados foram desenvolvidos considerando que cada polígono da COS tinha um ID único associado para a sua identificação. No entanto foram observadas algumas incoerências quando foi efetuada a verificação das áreas totais resultantes dos algoritmos (Figura 13) uma vez que estas áreas apresentaram um valor inferior ao valor total dos ficheiros originais. Devido a estas discrepâncias foi desenvolvido um algoritmo adicional (Figura 14) para a verificação dos indentificadores de cada polígono da COS.

```
#Criação de DataFrames para a verificação de áreas com identificadores iguais
ardidos_area1=COS2015.copy()
ardidos_area1.drop_duplicates(subset =["ID"], keep = False, inplace = True)

ardidos_area2=COS2015.copy()
ardidos_area2.drop_duplicates(subset =["ID","layer"], keep = False, inplace = True)

#Calculo do nº de áreas com identificadores iguais
N = len(ardidos_area1.ID)
NI = len(ardidos_area2.ID)
O = len(COS2015.ID)
print('Diferença (combinação ID, layer) : '+format(O-NI,'.0f'))
print('Diferença (ID) : '+format(O-N,'.0f'))

Diferença (combinação ID, layer) : 20
Diferença (ID) : 12889
```

Figura 14 - Algoritmo para a verificação de duplicados nos identificadores dos polígonos da COS

Ao observar os resultados do algoritmo da Figura 14 podemos verificar que dos 622629 polígonos da COS, 2.07% (12889) deles possuem pelo menos um ID duplicado. Após se ter observado este facto foi realizada uma tentativa de contornar o problema usando a combinação do ID do polígono com a região a que este pertence, infelizmente como se pode observar pelo algoritmo da Figura 14, mesmo usando a combinação ainda existiam 20 identificadores duplicados. Estes duplicados podem ser observados na Tabela 3 em combinações de 2.

Tabela 3 - Lista dos polígonos com a combinação de ID com Região (*layer*) iguais dos dados das interseções

ID	AREA	LAYER
19565	5,065703733	COS2015-V1-PT186_Alto_Alentejo
19565	1,34581E-06	COS2015-V1-PT186_Alto_Alentejo
52369	0,615996009	COS2015-V1-PT185_Leziria_Tejo
52369	0,70010725	COS2015-V1-PT185_Leziria_Tejo
86342	151,2675919	COS2015-V1-PT185_Leziria_Tejo
86342	0,70010725	COS2015-V1-PT185_Leziria_Tejo
126478	112,5942872	COS2015-V1-PT111_Alto_Minho
126478	4,961136481	COS2015-V1-PT111_Alto_Minho
152138	7,678970825	COS2015-V1-PT111_Alto_Minho
152138	4,961136481	COS2015-V1-PT111_Alto_Minho
211577	3,899792102	COS2015-V1-PT187_Alentejo_Central
211577	1,4736E-08	COS2015-V1-PT187_Alentejo_Central
215094	24,53040052	COS2015-V1-PT187_Alentejo_Central
215094	7,992E-08	COS2015-V1-PT187_Alentejo_Central
253470	18,73777139	COS2015-V1-PT186_Alto_Alentejo
253470	1,34581E-06	COS2015-V1-PT186_Alto_Alentejo
323847	600,4508078	COS2015-V1-PT187_Alentejo_Central
323847	7,992E-08	COS2015-V1-PT187_Alentejo_Central
329910	578,9651126	COS2015-V1-PT187_Alentejo_Central
329910	1,4736E-08	COS2015-V1-PT187_Alentejo_Central

Para resolver esta situação foi decidido criar um identificador fictício para substituir os ID presentes na COS. A este ID fictício deu-se o nome de COS_ID_FID, permitindo assim a utilização dos algoritmos como o da Figura 13. Com todos os problemas dos ficheiros resolvidos foi então possível executar os algoritmos para a criação das tabelas que seriam introduzidas na base de dados da plataforma web. Um exemplo destas tabelas pode ser encontrado na Tabela 4, enquanto as restantes podem ser encontradas no Anexo II.

Tabela 4 - Percentagens de área ardida em relação á área afetada de Mega Classe por ano

YEAR	2014	2015	2016	2018
AGRICULTURA	1,25316	1,55344	3,82659	26,48973
CORPOS_DE_ÁGUA	0,17195	0,70073	1,56187	1,721963
ESPAÇOS DESCOBERTOS OU COM VEGETAÇÃO ESPARSA	6,82906	26,0218	46,17268	35,82013
FLORESTAS	3,20691	7,19446	16,43363	20,03186
MATOS	3,76317	10,0640	20,61369	14,83498
PASTAGENS	4,42737	4,89244	6,48937	44,04852
SISTEMAS AGRO FLORESTAIS	2,58186	1,89703	3,83308	7,252381
TERRITÓRIOS ARTIFICIALIZADOS	0,35209	0,76384	2,71691	13,06419
ZONAS HÚMIDAS	0,00323	0,09664	0,05074	34,11948

4.2. Desenvolvimento do serviço para análise da informação dos incêndios

A segunda fase do desenvolvimento deste projeto foi focada no desenvolvimento do serviço da plataforma web usando a *framework* Django. O processo de desenvolvimento deste serviço foi dividido em 3 etapas diferentes, sendo estas etapas:

- A configuração das definições do Django;
- A criação dos modelos do Django e posterior inserção dos dados adquiridos na primeira fase do projeto;
- Desenvolvimento da *front-end*;

4.2.1. Definições do Django

Na primeira etapa do desenvolvimento do serviço web foi realizada a configuração das definições da *framework* Django. Foi durante esta configuração que o motor da base de dados foi escolhido e que foram instaladas as aplicações que ajudaram a tornar possível o desenvolvimento.

O motor da base de dados escolhido para este serviço da plataforma foi o SQLite3 v3.8.3. Este motor foi o escolhido de entre as várias opções que são suportadas pelo Django, sendo as outras opções, o PostgreSQL 9.4, o MySQL

5.6 e Oracle 12.1 [66]. O motor do SQLite3 foi o escolhido por ser um motor simples, que contém as funcionalidades necessárias para esta fase do projeto e que não necessita de nenhuma configuração prévia para funcionar com o Django.

Durante esta primeira etapa também foram efetuadas as instalações das aplicações suplementares ao Django que tornaram possível o desenvolvimento deste serviço. As aplicações instaladas foram o Django import-export e o Django graphos. Ambas as aplicações desempenharam um papel importante durante as etapas seguintes, tendo a aplicação Django import-export sido usada durante a segunda etapa e a aplicação Django graphos sido usada na terceira etapa.

4.2.2. Modelos do Django

Nesta segunda etapa do desenvolvimento foram criados os modelos no Django que foram usados para a criação das tabelas na base de dados. Foi também durante esta etapa que a aplicação Django import-export foi usada. A aplicação Django import-export é usada pelo Django para permitir a importação de dados guardados em ficheiros do tipo excel, csv, json, entre outros. Ao ser instalada esta aplicação adiciona estas funcionalidades à página de administração do Django.

A conceção dos modelos do Django foi a parte desta etapa que precisou de atenção especial, devido ao facto de ser uma das partes mais importantes desta segunda fase do desenvolvimento. A importância destes modelos vem do facto que é a partir destes modelos que o Django cria as tabelas na base de dados de forma a poderem ser usadas pela *framework*. Estes modelos são criados no Django dentro de um ficheiro especial do Django chamado `models.py` e é dentro deste ficheiro que são criados os vários modelos, estes são criados como classes de Python.

Os nomes dos modelos criados são os seguintes:

- `Data_mega_perc_country`;
- `Data_mega_perc`;
- `Data_sub_perc_country`;
- `Data_sub_perc`;
- `Data_reg_perc_country`;

- Data_reg_perc;
- Data_reg_mega_perc_14;
- Data_reg_mega_perc_15;
- Data_reg_mega_perc_16;
- Data_reg_mega_perc_17;
- Data_reg_mega_perc_18;
- Data_reg_mega_perc_country_14;
- Data_reg_mega_perc_country_15;
- Data_reg_mega_perc_country_16;
- Data_reg_mega_perc_country_17;
- Data_reg_mega_perc_country_18;
- Data_reg_sub_perc_14;
- Data_reg_sub_perc_15;
- Data_reg_sub_perc_16;
- Data_reg_sub_perc_17;
- Data_reg_sub_perc_18;
- Data_reg_sub_perc_country_14;
- Data_reg_sub_perc_country_15;
- Data_reg_sub_perc_country_16;
- Data_reg_sub_perc_country_17;
- Data_reg_sub_perc_country_18;

Os modelos que começam com Data_mega, Data_sub e Data_reg são os modelos que contêm os dados das percentagens de área ardida em relação à área do país (perc_country) e em relação à área afetada (perc) dos anos em função das mega classes, classes e regiões respetivamente. Os modelos que começam com Data_reg_mega e Data_reg_sub são os modelos que contêm os dados das regiões em função tanto das mega classes como das classes. Estes tiveram de ser separados pelos diferentes anos devido à forma como a aplicação Django graphos lê os dados a partir da base de dados.

Na Figura 15 pode ser observado o modelo que foi usado para guardar os dados da Tabela 4. Como se pode observar na figura os nomes de cada coluna da base de dados e o tipo de dados correspondente é igual ao que se encontra na tabela.

```

class Data mega_perc(models.Model):
    year = models.PositiveSmallIntegerField(null=True)
    agricultura = models.FloatField(null=True)
    corpos_de_água = models.FloatField(null=True)
    espaços_descobertos_ou_com_vegetação_esparça = models.FloatField(null=True)
    florestas = models.FloatField(null=True)
    matos = models.FloatField(null=True)
    pastagens = models.FloatField(null=True)
    sistemas_agro_florestais = models.FloatField(null=True)
    territórios_artificializados = models.FloatField(null=True)
    zonas_húmidas = models.FloatField(null=True)

```

Figura 15 - Modelo do Django usado para a inserção dos dados da Tabela 4

O requisito dos nomes do modelo terem de ser iguais aos nomes na tabela vem da aplicação Django import-export, que necessita deste requisito para conseguir detetar e adicionar os dados dos ficheiros produzidos na primeira fase do projeto à base de dados. A utilização desta aplicação foi necessária de forma a ser possível otimizar o processo de importação dos dados para a base de dados.

4.2.3. Desenvolvimento da *front-end*

Nesta terceira e última etapa do desenvolvimento do serviço foi realizado o desenvolvimento da *front-end* do serviço, onde se transformaram os dados presentes na base de dados em gráficos preparados para serem analisados pelos utilizadores do serviço. Nesta etapa do desenvolvimento a aplicação *graphos* foi usada para apresentar gráficos de barras provenientes da Google Charts API.

O desenvolvimento da *front-end* começou pela criação dos ficheiros html que contêm tanto a informação da apresentação da página *web* do serviço como alguns comandos específicos do Django que são interpretados pelo próprio. Estes ficheiros html são usados nas *views* do Django, estas recebem e transmitem informação para esses ficheiros através dos comandos.

Nesta etapa foi desenvolvida uma *view* diferente para cada página do serviço. Em cada uma das *views* desenvolvidas 2 elementos estavam sempre presentes, o formulário e a aplicação Django *graphos*. As *views* criadas para esta fase do projeto funcionam todas de forma semelhante, mudando apenas o formulário associado e os dados que são retirados da base de dados.

O funcionamento das *views* começa pela exibição do formulário na página do serviço seguido da verificação de se o mesmo já foi preenchido. No caso do formulário já ter sido preenchido, é dentro da mesma *view* que os dados do formulário são analisados e dependentemente dos mesmos, são efetuadas as *queries* à base de dados necessárias para adquirir os dados solicitados. Estes dados são então transformados pela aplicação *graphos* para o formato que permite a sua apresentação em gráficos. A transmissão dos dados e a comunicação das *views* com o código nos ficheiros *html* é efetuada com recurso à linguagem de programação específica do Django. Nas Figuras 16 e 18 pode ser observado um exemplo de uma das *views* e o ficheiro *html* com o qual a *view* está a comunicar. O resultado da comunicação entre os códigos presentes nas Figuras 16,18 pode ser observado na Figura 18 (formulário).

```
def mega_class_page(request):
    form_mega= megaform()
    if request.method=='POST':
        form1 = megaform(request.POST)
        if form1.is_valid():
            year = form1.cleaned_data.get('Ano')
            classe = form1.cleaned_data.get('MegaClasses')
            perc = form1.cleaned_data.get('Percentagem')
            if perc == 'perc_country':
                queryset = Data_mega_perc_country.objects.filter(year__in=year)
            elif perc == 'perc_burned':
                queryset = Data_mega_perc.objects.filter(year__in=year)
            classe.insert(0, 'year')
            data_source = ModelDataSource(queryset, fields=classe)
            # Chart object
            chart = ColumnChart(data_source, width=1000, height=500,
                                options={'title': f"Data {year}",
                                         'vAxis': {'title': "Percentagem Área Ardida"}})
            context = {'megaform':form_mega, 'chart': chart}
            return render(request, 'mega_year.html', context)
        context={'megaform':form_mega}
        return render(request, 'mega_year.html', context)
```

Figura 16 - Código da *view* da página dos gráficos de Mega Classe/Ano

IPSTERS [Início](#) [Mega Classe](#) [Classe](#)

Gráficos de Mega Classes/Ano

Ano:

- 2014
- 2015
- 2016
- 2018

MegaClasses:

- Matos
- Florestas
- Agricultura
- Corpos de água
- Espaços descobertos ou com vegetação esparsa
- Pastagens
- Sistemas agro-florestais
- Territórios artificializados
- Zonas húmidas

Porcentagem:

- Porcentagem da área ardida em relação à área do país
- Porcentagem da área ardida em relação à área afetada

Figura 18 - Página dos gráficos de Mega Classe/Ano, mostrando o formulário

```
{% extends "base.html" %}

{% block content %}

<h3 align="center">Gráficos de Mega Classes/Ano</h3>

<form action="/Mega_Year" method="POST">
  {% csrf_token %}
  {{megaform}}
  <input type="submit" value="submit">
</form>
<div align="center">
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
  google.load("visualization", "1", {packages:["corechart"]});
</script>
{{ chart.as_html }}
</div>

{% endblock %}
```

Figura 17 - Código html da página dos gráficos de Mega Classe/Ano

4.3. Desenvolvimento do serviço para visualização dos mapas

Na fase final deste projeto foi desenvolvido o serviço da plataforma web que permite aos utilizadores visualizar os mapas que contêm a informação sobre a localização das áreas afetadas pelos incêndios desde 2014 até 2018. Este serviço foi desenvolvido com recurso ao Django sendo que neste serviço foi utilizado o pacote bibliotecas do leaflet e o módulo GeoDjango para tornar possível a visualização dos mapas e aceder às capacidades geográficas da base de dados. O desenvolvimento desta última fase pode ser dividido em 3 etapas diferentes, sendo estas:

- A configuração do Django e da base de dados;
- A criação dos modelos GeoDjango e a inserção dos dados geográficos na base de dados;
- O desenvolvimento da *front-end*;

4.3.1. Configuração do Django e da base de dados

Nesta primeira etapa desta fase foi realizada a configuração do Django e da base de dados. Esta diferencia-se da configuração efetuada durante a segunda fase devido á necessidade de utilizar o módulo GeoDjango, o que veio alterar tanto a escolha do motor da base de dados como das aplicações suplementares necessárias.

Em relação ao motor da base de dados nesta fase foi utilizado o PostgreSQL 9.4, esta mudança deveu-se ao facto de esta ser uma das bases de dados suportada pelo GeoDjango e por estar mais bem equipada para o processamento da informação geográfica que o motor utilizado durante a segunda fase. Devido à necessidade de utilizar o GeoDjango para trabalhar com dados geográficos no Django, não foi possível utilizar este novo motor da base de dados sem antes terem sido instalados alguns requisitos como as bibliotecas GEOS, GDAL, PROJ.4, PostGIS e o adaptador psycopg2 [45, 67]. Entre os requisitos necessários o PostGIS, acaba por ser o mais importante, por ser aquele que vai transformar a base de dados PostgreSQL numa base de dados espacial com a capacidade de trabalhar com dados espaciais. No entanto, o PostGIS precisa do

adaptador psychog para ser possível a comunicação entre o GeoDjango e o PostGIS, este é um adaptador de Python para a PostgreSQL [67, 68].

Durante esta etapa ainda foi necessário recorrer à instalação da aplicação suplementar Django Leaflet. Esta aplicação foi necessária de forma a permitir a visualização da informação geográfica presente na base de dados num mapa interativo disponível para todos os utilizadores deste serviço.

4.3.2. Modelos GeoDjango

Nesta segunda etapa do desenvolvimento foram criados os modelos GeoDjango que foram utilizados para guardar os dados geográficos provenientes dos ficheiros do tipo .shp (*ShapeFiles*). Estes modelos diferem dos modelos normalmente utilizados no Django pela possibilidade de serem criados com campos de informação geográfica, enquanto os modelos do Django não possuem essa opção. Além desta diferença, a forma como cada modelo é criado também difere de um modelo Django normal pelo facto dos seus campos terem de ser iguais em nome e tipo aos dados existentes nos *ShapeFiles*. No entanto o GDAL dispõe da ferramenta ogrinspect que permite a leitura dos *ShapeFiles* e permite ainda gerar automaticamente os modelos necessários. Na Figura 19 pode ser observada a utilização desta ferramenta na criação automática do modelo usado para guardar a informação do mapa de incêndios de 2018.

```
(base) PS D:\Universidade\TESE\Django\geodjango> python manage.py ogrinspect World\data\AA_2018_23112018.shp Fires2018 --srid=3763 --multi
# This is an auto-generated Django model module created by ogrinspect.
from django.contrib.gis.db import models

class Fires2018(models.Model):
    id = models.IntegerField()
    country = models.CharField(max_length=4)
    countryful = models.CharField(max_length=200)
    province = models.CharField(max_length=120)
    commune = models.CharField(max_length=100)
    firedate = models.CharField(max_length=20)
    area_ha = models.IntegerField()
    lastupdate = models.CharField(max_length=20)
    geom = models.MultiPolygonField(srid=3763)
```

Figura 19 - Utilização da ferramenta Ogrinspect para gerar o modelo onde será guardada a informação do mapa de 2018

Após a criação dos 5 modelos para os anos de 2014, 2015, 2016, 2017 e 2018 chamados respetivamente Fires2014, Fires2015, Fires2016, Fires2017 e Fires2018, foi necessário encontrar uma forma de guardar os dados existentes

nos ficheiros *ShapeFile* nas tabelas da base de dados correspondentes. Para conseguir alcançar este objetivo recorreu-se novamente a uma ferramenta do GDAL. Neste caso a ferramenta utilizada foi o `ogr2ogr`, que tem o objetivo de converter dados de recursos simples entre vários formatos. Nesta etapa, esta ferramenta foi utilizada na conversão dos dados dos ficheiros *ShapeFiles* em dados reconhecidos pela base de dados PostgreSQL, sendo estes dados automaticamente adicionados à base de dados pela mesma ferramenta. Como exemplo de utilização desta ferramenta este foi o comando utilizado usando a linha de comandos para transferir os dados do mapa dos incêndios de 2018:

```
ogr2ogr -overwrite -f "PostgreSQL" PG:"dbname=geo user=postgres password=admin" AreasArdidas_2017_031002018_ETRS89PTTM06.shp -nlt PROMOTE_TO_MULTI -nln World_fires2017 -append
```

No comando em cima apresentado podem ser destacadas as várias opções utilizadas como:

- `-overwrite` – apaga a tabela e recria a sem quaisquer dados;
- `-f` – onde é especificado as informações da base de dados seguidas do nome do ficheiro do tipo *ShapeFile*;
- `-nlt` – define o tipo de geografia a ser introduzido na tabela;
- `-nln` – define o nome da tabela onde os dados vão ser introduzidos;
- `-append` – faz com que a ferramenta introduza os dados numa tabela já existente;

4.3.3. Desenvolvimento da *front-end*

Nesta etapa final do desenvolvimento do serviço da plataforma *web* foi efetuado o desenvolvimento da *front-end* do serviço onde ocorreu a transformação dos dados geográficos presentes na base de dados, num formato reconhecido pela aplicação Django leaflet, a serem observados no mapa interativo disponibilizado pela mesma aplicação. O desenvolvimento da *front-end* deste serviço começou pela criação dos ficheiros HTML responsáveis tanto pelo aspeto para o utilizador, como pela receção dos dados e posterior disponibilização dos mesmo num mapa interativo. Estes ficheiros ao contrário dos ficheiros do serviço anteriormente desenvolvido, não recebem os dados diretamente das views mas sim a partir das respostas em Http provenientes das mesmas. Esta mudança de funcionamento vem do facto que a aplicação Django leaflet necessita dos dados

geográficos no formato *geojson*. Este é um formato que serve para codificar uma variedade de estruturas de dados geográficos entre elas a estrutura *MultiPolygon* usada neste projeto [69]. É devido a esta necessidade de a aplicação usar os dados geográficos no formato *geojson* que se torna necessário realizar a transformação dos dados geográficos presentes na base de dados para o formato *geojson*, esta transformação é realizada dentro das *views* e é comunicada à aplicação através de uma resposta em Http.

Para as *views* utilizadas alcançarem o objetivo de transformar os dados geográficos disponíveis na base de dados em dados no formato *geojson* é utilizada uma ferramenta do Django chamada *serialize*. Esta ferramenta fornece ao Django um mecanismo simples para transformar os dados guardados pelos modelos do Django na base de dados em dados do tipo xml,yaml,json e geojson [70].

A *front-end* deste serviço está desenvolvida para funcionar de uma forma automatizada na medida em que o utilizador necessita apenas de introduzir o url do ano que quer visualizar para lhe ser disponibilizado o mapa com a informação da localização dos incêndios. A comunicação entre as restantes *views* e url é efetuada em segundo plano. Esta comunicação ocorre sempre por uma ordem específica sendo que começa pela comunicação do url do ano escolhido pelo utilizador usando um menu com uma *view* (ver Figura 20) que apenas redireciona esse pedido para o ficheiro HTML que contém o código da aplicação Django leaflet (Figura 21). É dentro deste ficheiro que existe um script escrito em javascript que serve para juntar os dados disponibilizados com o mapa. É este script que envia um pedido de informações a um novo url. Este novo url realiza a comunicação com a *view* (Figura 20) onde está inserida a ferramenta de serialização do Django para efetuar a transformação dos dados e enviar os mesmos para o novo url. São estes dados que vão ser lidos pelo script utilizado no ficheiro HTML onde se encontra o código da aplicação Django leaflet. Depois dessa leitura ser efetuada é então apresentado ao utilizador do serviço o mapa com os polígonos das áreas afetadas pelos incêndios.

```
#####
# Views do mapa de 2014

#serialização dos dados
def fogos_2014_data(request):
    fire_as_geojson = serialize('geojson', Fires2014.objects.all(), fields=__('geom'))
    return HttpResponse(fire_as_geojson, content_type='json')

#redirecionar o url para o ficheiro html
class fogos_2014(TemplateView):
    template_name = 'world/fogos2014-mapa.html'
```

Figura 20 - Código da view de serialização (fogos_2014_data) e da view de redirecionamento (fogos_2014) do ano de 2014

```
{% extends "base.html" %}

{% block content %}

{% load leaflet_tags %}
{% load static %}
<head>
  <style media="screen">
    #fire { width:100%; height:100% }
  </style>
  {% leaflet_js %}
  {% leaflet_css %}
  <script type="text/javascript" src="{% static 'leaflet-ajax/dist/leaflet.ajax.js' %}"></script>
</head>

<body>

<script type="text/javascript">
  function map_init_basic (map, options) {
    var dataset = new L.GeoJSON.AJAX("{% url 'fire2014' %}", { });
    dataset.addTo(map);
  }
</script>
  {% leaflet_map "fire" callback="window.map_init_basic" %}
</body>

{% endblock %}
```

Figura 21 - Código html da apresentação do mapa de 2014

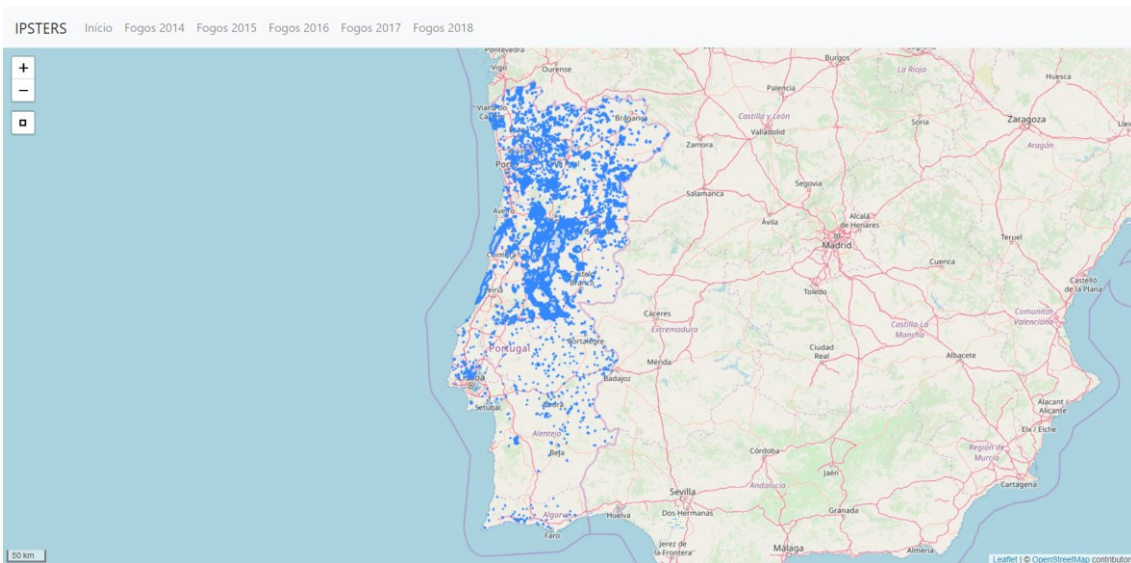


Figura 22 - Exemplo do serviço de visualização dos mapas, apresentando os dados do ano de 2017

5

Resultados

Neste capítulo são apresentados os resultados dos dois serviços da plataforma desenvolvida separados em dois subcapítulos: o subcapítulo da informação dos incêndios onde estão disponíveis os gráficos disponíveis pelo serviço de análise da informação dos incêndios e um subcapítulo onde estão disponíveis as imagens dos mapas dos anos de 2014 a 2018 disponibilizados pelo serviço de visualização dos mapas.

5.1. Informação dos incêndios

Neste subcapítulo é apresentada toda a *interface* do serviço de análise da informação dos incêndios e serão também apresentados exemplos dos vários gráficos disponíveis para os utilizadores. Começando pelo menu que se encontra disponível para os utilizadores, este pode ser observado na Figura 23.

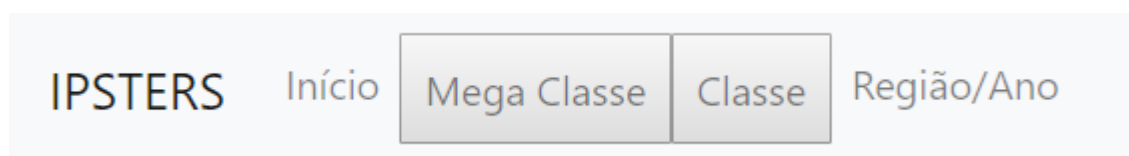


Figura 23 - Menu do serviço de análise da informação dos incêndios

Como se pode observar na Figura 23, o menu desta aplicação oferece ao utilizador uma escolha de 4 opções das quais as opções Mega Classe e Classe funcionam como uma “*dropbox*” apresentando cada uma destas 2 opções de escolha, deixando assim ao utilizador um total de 6 páginas diferentes a selecionar. Cada uma destas 6 páginas tem uma função diferente, sendo que 5 delas se destinam à apresentação dos diversos gráficos possíveis e uma sexta página que se dedica a dar as boas vindas ao utilizador. Esta última página é acedida através da opção “Início”. As restantes opções são respetivamente:

- Mega Classe/Ano – Gráficos de mega classes por ano;
- Mega Classe/Região – Gráficos de mega classes por regiões;
- Classe/Ano – Gráficos de classes por ano;
- Classe/Região – Gráficos de classes por regiões;
- Região/Ano – Gráficos de Regiões por ano;

Embora estas 5 páginas tenham umas funções diferentes umas das outras, a forma de como o utilizador escolhe a informação que pretende visualizar é através de formulários que estão presentes nestas 5 páginas de uma forma idêntica. Com isto quer dizer-se que o formulário para a escolha do ano é idêntico em todas as páginas que necessitem da informação do ano. O mesmo se pode dizer para o formulário das mega classes, classes, regiões e do tipo de percentagem que se pretende visualizar.

Ano:

- 2014
- 2015
- 2016
- 2017
- 2018

Figura 24 - Formulário para escolha do ano/anos para os quais se pretenda visualizar a informação

Como se pode observar na Figura 24, o formulário de escolha dos anos é um formulário simples que oferece ao utilizador a opção de escolher o ano ou anos dos quais este pretenda visualizar a informação. Este formulário funciona utilizando caixas de seleção para permitir a seleção de múltiplos anos.

Megaclases:

- Matos
- Florestas
- Agricultura
- Corpos de água
- Espaços descobertos ou com vegetação esparsa
- Pastagens
- Sistemas agro-florestais
- Territórios artificializados
- Zonas húmidas

Figura 25 - Formulário para escolha da Mega classe(s) para as quais se pretenda visualizar a informação

Na Figura 25 encontra-se o formulário para a escolha da(s) mega classe(s) que se pretenda visualizar. A escolha para estes nomes de mega classes vem das classes de nível 1 da COS. Este formulário, tal como o formulário da Figura 24 funciona através de caixas de seleção para permitir ao utilizador uma maior liberdade de escolha na informação que pretenda visualizar no gráfico

Classes:

- Tecido urbano contínuo
- Tecido urbano descontínuo
- Indústria comércio e equipamentos gerais
- Redes viárias e ferroviárias e espaços associados
- Áreas portuárias
- Aeroportos e aeródromos
- Áreas de extracção de inertes
- Áreas de deposição de resíduos
- Áreas em construção
- Espaços verdes urbanos
- Campos de golfe
- Outras instalações desportivas e equipamentos de lazer
- Equipamentos culturais e outros e zonas históricas
- Culturas temporárias de sequeiro e regadio
- Vinhas
- Pomares
- Olivais
- Pastagens permanentes
- Culturas temporárias e pastagens associadas a culturas permanentes
- Sistemas culturais e parcelares complexos
- Agricultura com espaços naturais e semi naturais
- Sistemas agro florestais de outros carvalhos
- Florestas de sobreiro
- Florestas de outros carvalhos
- Florestas de castanheiro
- Florestas de eucalipto
- Florestas de espécies invasoras
- Florestas de outras folhosas
- Florestas de pinheiro bravo
- Florestas de pinheiro manso
- Florestas de outras resinosas
- Vegetação herbácea natural
- Matos
- Espaços descobertos ou com pouca vegetação
- Zonas húmidas
- Cursos de água
- Planos de água
- Desembocaduras fluviais
- Oceano
- Sistemas agro florestais de pinheiro manso
- Florestas de azinheira
- Sistemas agro florestais de outras espécies
- Lagoas costeiras
- Sistemas agro florestais de sobreiro
- Sistemas agro florestais de azinheira
- Sistemas agro florestais de outras misturas
- sistemas_agro_florestais_de_sobreiro_com_azinheira
- arrozais

Figura 26 - Formulário para escolha da Classe(s) para as quais se pretenda visualizar a informação

Na Figura 26 pode ser observado o formulário disponibilizado aos utilizadores para a seleção da classe ou classes que o utilizador pretenda visualizar a informação no gráfico. Este formulário acabou por ser o maior de todos por conter todas as 48 classes que pertencem ao nível 5 da COS. O funcionamento deste formulário é idêntico aos anteriores fazendo também uso de caixas de seleção.

Regioes:

- Alto Minho
- Cavado
- Ave
- Area Metropolitana do Porto
- Alto Tamega
- Tamega e Sousa
- Douro
- Terras de Tras os Montes
- Algarve
- Oeste
- Regiao de Aveiro
- Regiao de Coimbra
- Regiao de Leiria
- Viseu dao Lafoes
- Beira Baixa
- Medio Tejo
- Beiras Serra Estrela
- Area Metropolitana de Lisboa
- Alentejo Litoral
- Baixo Alentejo
- Leziria Tejo
- Alto Alentejo
- Alentejo Central

Figura 27 - Formulário para escolha das Regiões para as quais se pretenda visualizar a informação

Na Figura 27 pode ser observado o formulário para a escolha das regiões de Portugal. No entanto, estas regiões são as que a COS utiliza sendo estas as regiões do NUTS III. Este formulário tal como os anteriores funciona com a utilização de caixas de seleção para permitir ao utilizador uma maior liberdade na escolha da informação pretendida.

Percentagem:

- Percentagem da área ardida em relação á área do país
- Percentagem da área ardida em relação á área afetada

Figura 28 - Formulário para a escolha do tipo de percentagem que se deseja visualizar

Na Figura 28 pode ser observado o último formulário que é disponibilizado aos utilizadores. Este tem o objetivo de oferecer ao utilizador a escolha entre dois tipos diferentes de percentagens: a percentagem da área ardida em relação á área do país e a percentagem de área ardida em relação á área afetada. Esta última refere-se à área dos polígonos afetados pelos incêndios. Este formulário ao

contrário dos anteriores apenas permite que se escolha uma das duas opções disponíveis.

Os resultados dos formulários apresentados nas Figura 24-28 resultam sempre em um ou mais gráficos dependendo da página e das opções escolhidas. Nas Figuras 29-33 podem ser observados exemplos de gráficos produzidos por cada uma das páginas. Devido ao grande número de combinações possíveis e de opções entre as 5 páginas para os gráficos não é possível apresentar em imagens todos estes gráficos pelo que serão apenas apresentados os gráficos com a combinação de todas as opções a serem apresentadas e apresentando a percentagem de área ardida em relação à área do país.

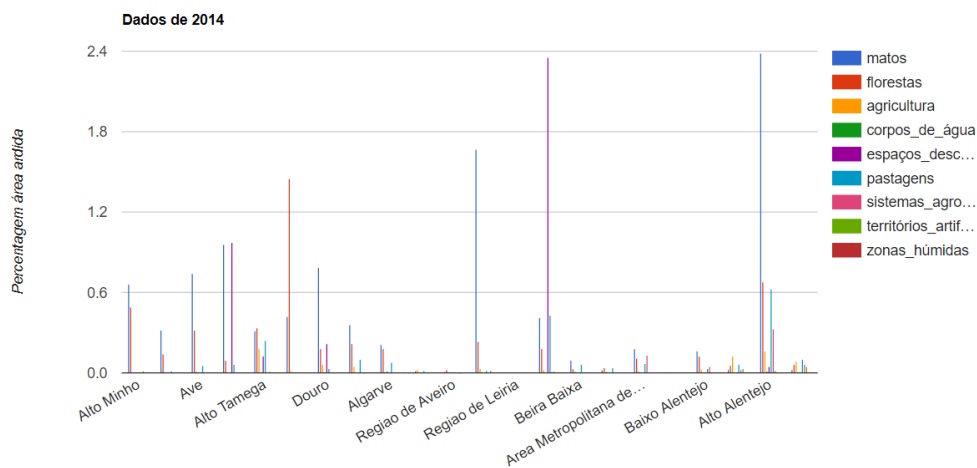


Figura 29 - Gráfico da página de Mega classe/Região contendo todas as opções selecionadas

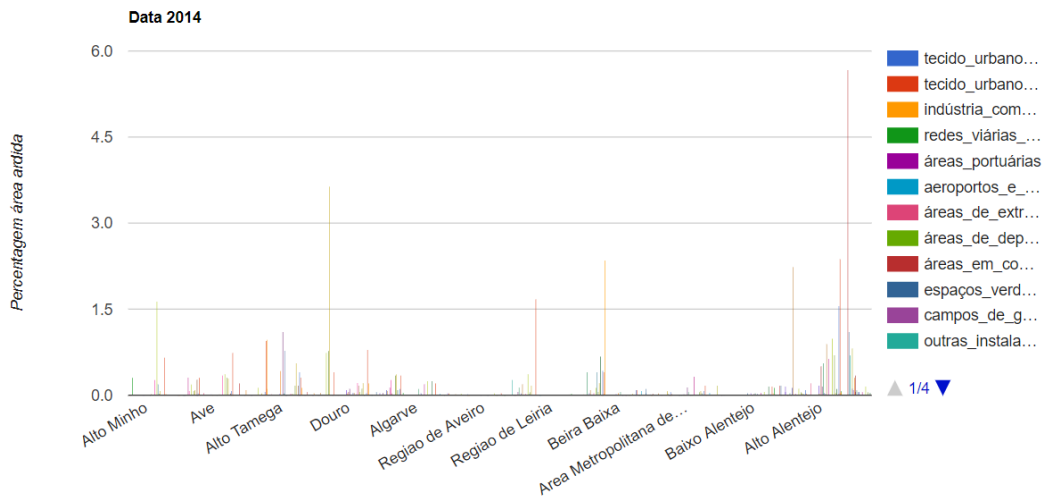


Figura 30 - Gráfico da página de Classe/Região contendo todas as opções selecionadas

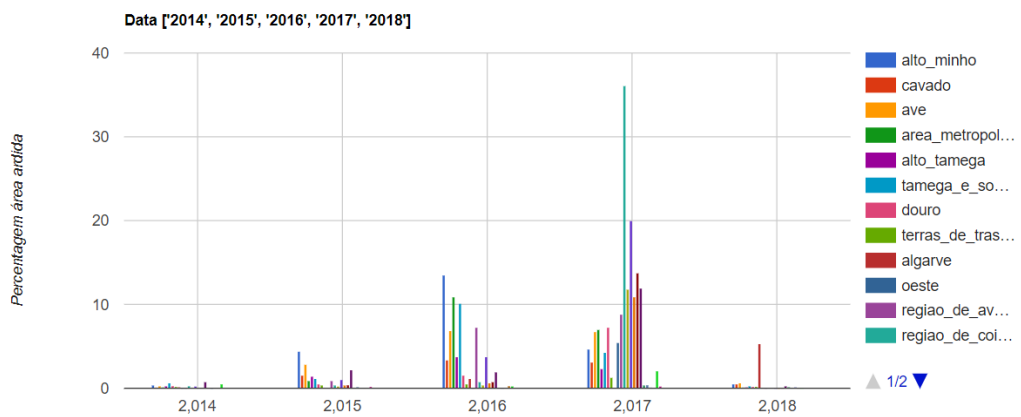


Figura 31 - Gráfico da página de Regiões/Ano contendo todas as opções selecionadas

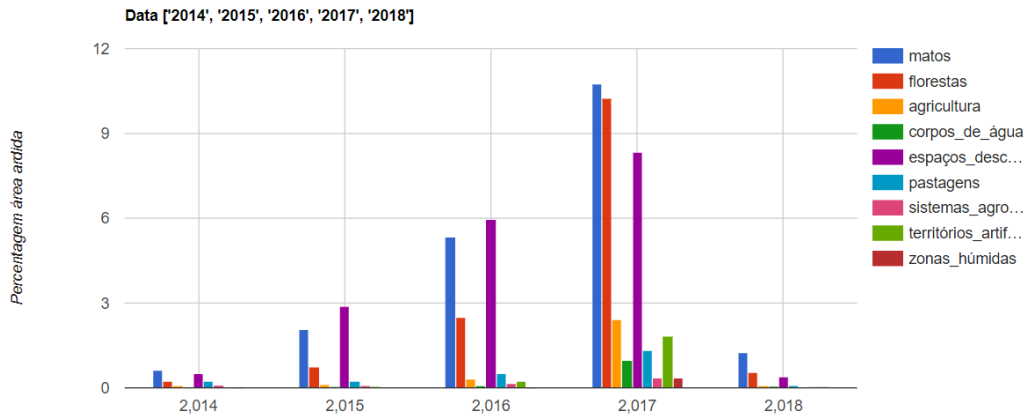


Figura 32 - Gráfico da página de MegaClasse/Ano contendo todas as opções selecionadas

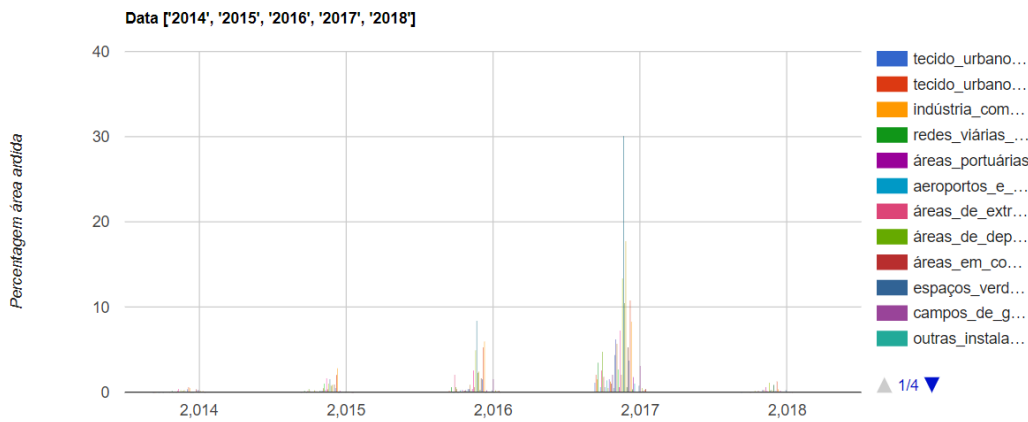


Figura 33 - Gráfico da página de Classe/Ano contendo todas as opções selecionadas

5.2. Serviço de visualização dos mapas dos incêndios

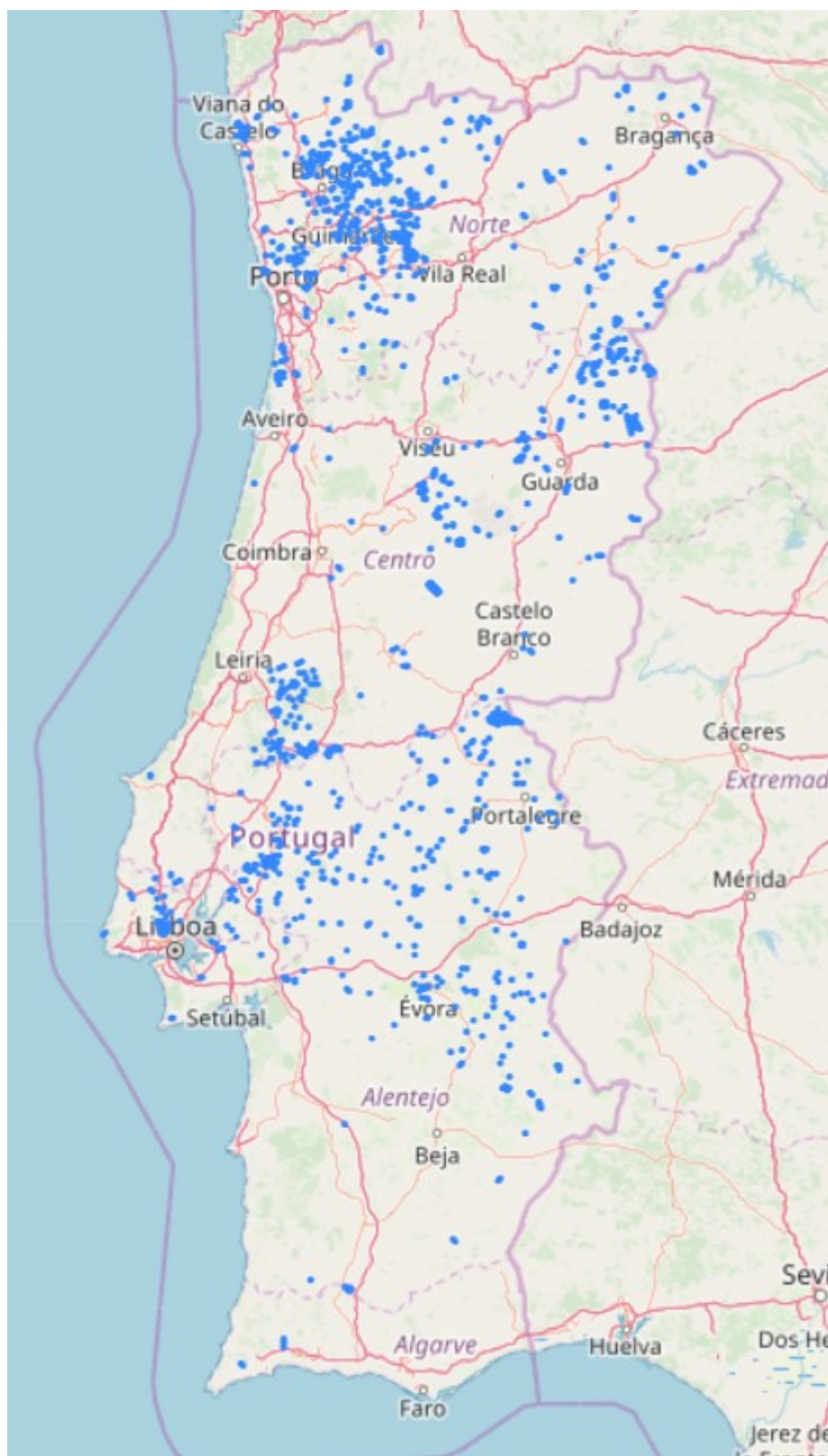


Figura 34 - Mapa com a localização das áreas afetadas pelos incêndios de 2014

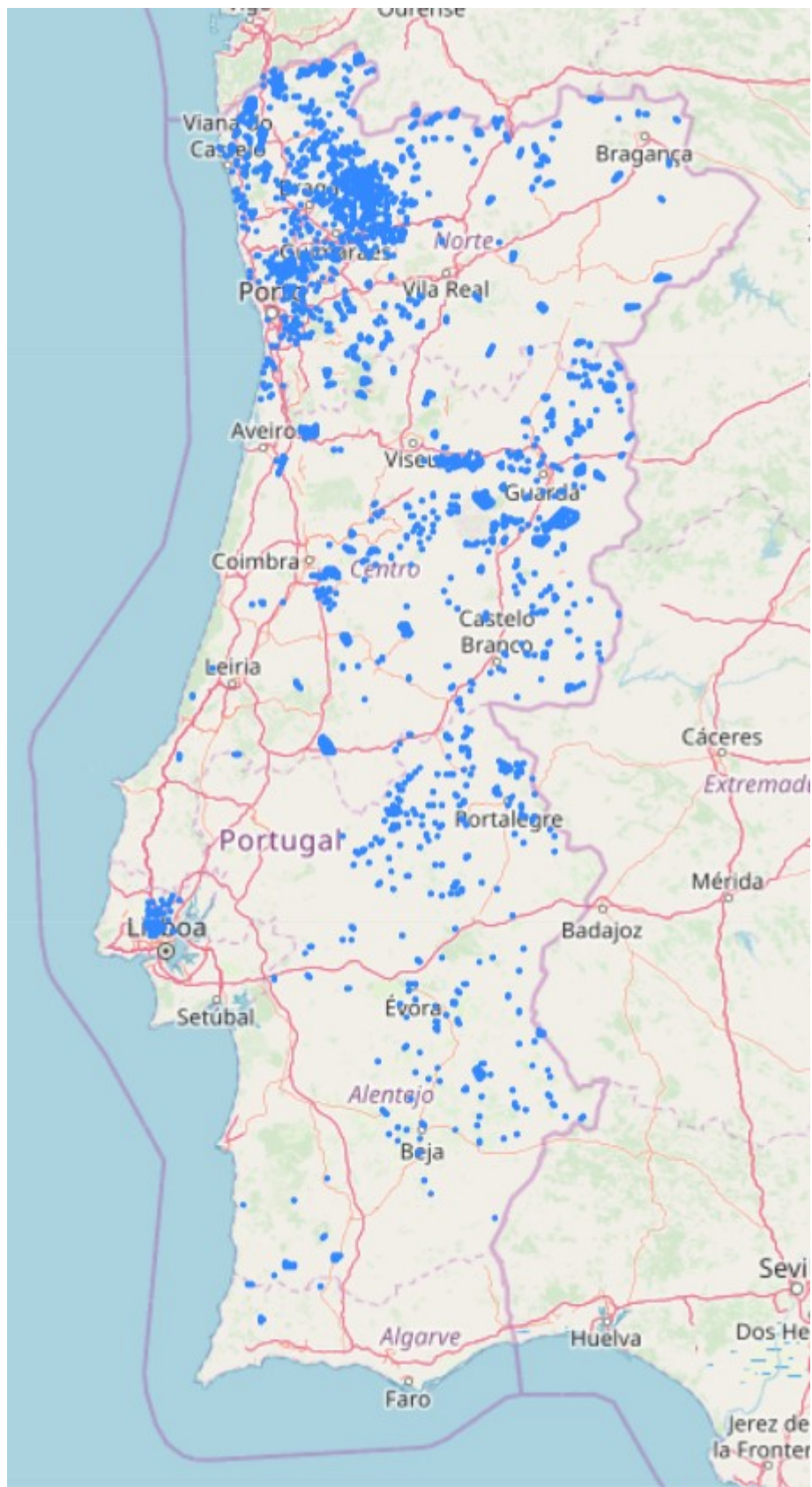


Figura 35 - Mapa com a localização das áreas afetadas pelos incêndios de 2015

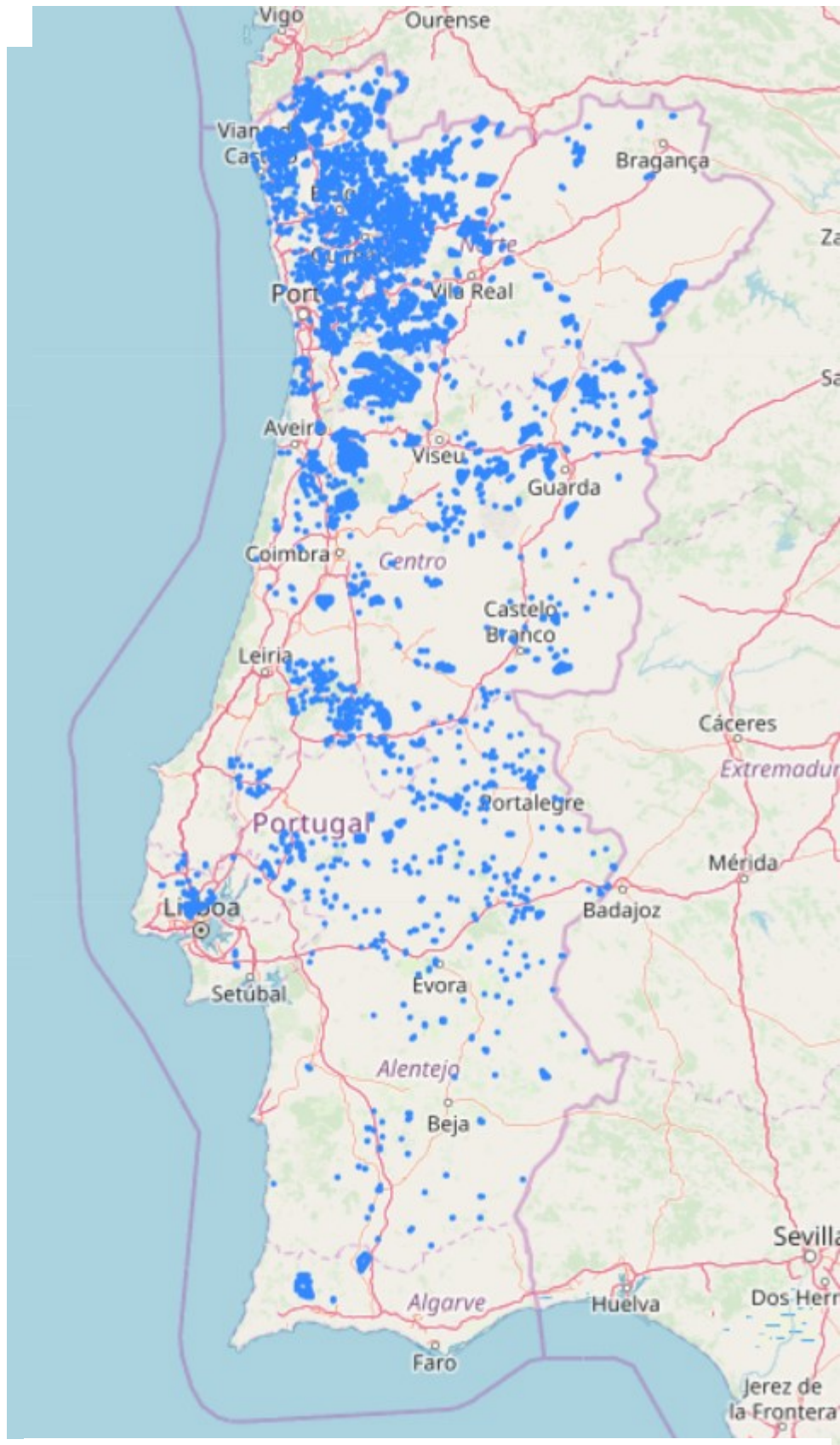


Figura 37 - Mapa com a localização das áreas afetadas pelos incêndios de 2017

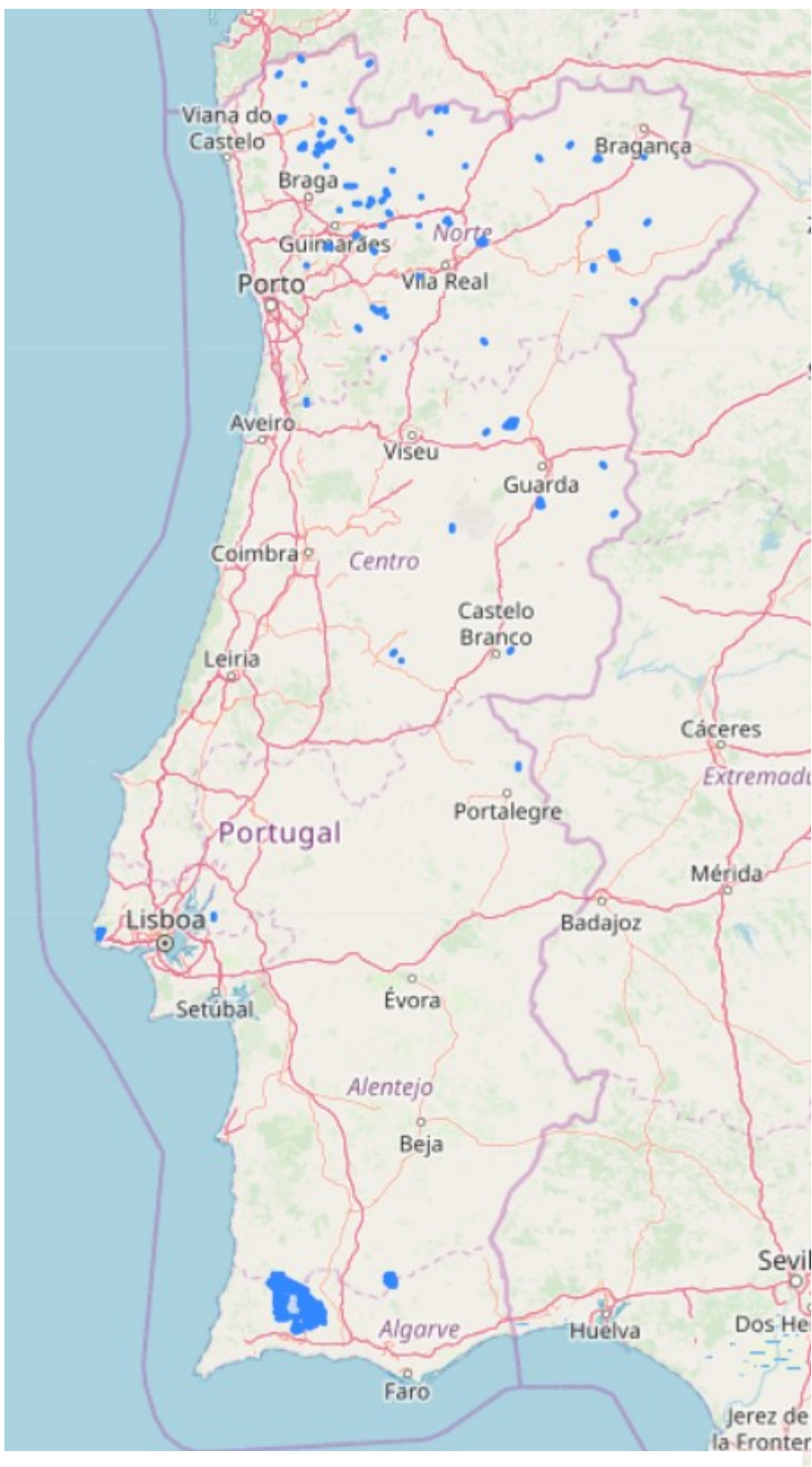


Figura 38 - Mapa com a localização das áreas afetadas pelos incêndios de 2018

Nas figuras 34-38 podem ser observados os mapas dos incêndios de 2014 até 2018. Na forma em que são expostos pelo serviço da plataforma ao utilizador, é preciso ainda que notar que este serviço permite ao utilizador um certo nível de interatividade com estes mapas. O nível de interatividade que é disponibilizado ao utilizador é a capacidade de ampliar e reduzir o zoom e de arrastar o mapa o que permite ao utilizador observar qualquer área de Portugal que este deseje. O nível de zoom disponível tem uma resolução por unidade de área de 10m a 5000km. Em relação ao desempenho este serviço ficou com um desempenho inferior ao esperado, sendo que mesmo com os mapas a serem apresentados praticamente instantaneamente a camada que contém os polígonos dos incêndios de cada ano pode demorar de 10s (mapa de 2014) a 1 minuto (mapa de 2017) a apresentar os dados. Esta diferença vem da necessidade da plataforma fazer o carregamento de toda a informação o que faz com que anos como 2017, que contêm muitos dados a serem carregados demorem consideravelmente mais que anos onde existam menos dados como o ano de 2014.



Conclusões

6.1. Objetivos cumpridos

Os objetivos propostos para este projeto foram cumpridos na sua totalidade. Durante a fase do tratamento de dados foi onde ocorreram os maiores contratempos devido a problemas nos identificadores dos polígonos. Estes contratempos foram resolvidos usando uma abordagem diferente com a criação de novos identificadores o que fez com que esta fase do projeto tenha sido ultrapassada com sucesso.

Em relação ao desenvolvimento do serviço para a visualização dos dados estatísticos este pode ser considerado completo com o desenvolvimento apresentado no capítulo 4.2. Foi desenvolvido um serviço que oferece ao utilizador uma variedade de opções que permitem a combinação de dados de mega classes, classes, regiões e anos em gráficos que possibilitam uma análise mais gráfica e compacta dos dados da interseção dos incêndios com os dados da COS2015.

O desenvolvimento do serviço para a visualização dos mapas dos incêndios foi o último objetivo proposto. Este foi também alcançado com o desenvolvimento descrito no capítulo 4.3. Este serviço foi desenvolvido de forma a oferecer ao utilizador um menu simples onde é possível selecionar o ano em que deseja observar a localização das áreas afetadas pelos incêndios. Devido à grande quantidade de informação processada por esta plataforma, o seu desempenho

acabou por ser inferior ao esperado uma vez que esta demora algum tempo até apresentar os dados pedidos.

6.2. Trabalho futuro

Com o desenvolvimento desta plataforma completo, surgem várias possibilidades para serem consideradas no que respeita ao trabalho que poderá vir a ser desenvolvido utilizando esta plataforma. Entre as possibilidades de trabalho a ser realizado no futuro, uma delas seria uma análise profunda tanto qualitativa como quantitativa dos dados presentes na plataforma. Com este estudo será certamente possível chegar a conclusões sobre a probabilidade de que cada área vir a arder caso seja afetada por um incêndio. Outra possibilidade para um trabalho futuro envolvendo a plataforma desenvolvida durante este projeto seria a passagem desta plataforma de um servidor local para um servidor público o que permitiria o acesso à mesma por um número maior de pessoas. Com a passagem desta plataforma para um servidor público propunha com que esta plataforma fosse disponibilizada de forma gratuita para todos e que o seu código fosse igualmente disponibilizado ao público no formato de *open source*.

Referências

- [1] “About Copernicus | Copernicus,” [Online]. Available: <https://www.copernicus.eu/en/about-copernicus>. [Acedido em 22 Julho 2019].
- [2] “CA3 - Computational Intelligence Research Group,” [Online]. Available: https://www.ca3-uninova.org/project_ipsters. [Acedido em 29 Agosto 2019].
- [3] “CORINE Land Cover — Copernicus Land Monitoring Service,” [Online]. Available: <https://land.copernicus.eu/pan-european/corine-land-cover>. [Acedido em 09 Agosto 2019].
- [4] “DGTerritório - Carta de Uso e Ocupação do Solo de Portugal Continental (COS),” [Online]. Available: http://www.dgterritorio.pt/dados_abertos/cos/. [Acedido em 09 Agosto 2019].
- [5] “Saber mais sobre SNIG | SNIG,” [Online]. Available: <https://snig.dgterritorio.gov.pt/saber-mais>. [Acedido em 09 Agosto 2019].
- [6] “Python Data Analysis Library — pandas: Python Data Analysis Library,” [Online]. Available: <https://pandas.pydata.org/>. [Acedido em 20 Setembro 2019].
- [7] “FIRMS,” [Online]. Available: <https://firms.modaps.eosdis.nasa.gov/>. [Acedido em 18 Julho 2019].
- [8] “What is MODIS? | Center for Earth Observation,” [Online]. Available: <https://yceo.yale.edu/what-modis>. [Acedido em 18 Julho 2019].
- [9] “MODIS WEB,” [Online]. Available: <https://modis.gsfc.nasa.gov/about/>. [Acedido em 18 Julho 2019].
- [10] W. Schroeder e L. Giglio, NASA VIIRS Land Science Investigator Processing System (SIPS) Visible Infrared Imaging Radiometer Suite

(VIIRS) 375 m & 750 m Active Fire Products - Product User's Guide Version 1.4, NASA, 2018.

[11] "Copernicus In Brief | Copernicus," [Online]. Available: <https://www.copernicus.eu/en/about-copernicus/copernicus-brief>. [Acedido em 22 Julho 2019].

[12] "Emergency | Copernicus," [Online]. Available: <https://www.copernicus.eu/en/services/emergency>. [Acedido em 22 Julho 2019].

[13] EFFIS, "EFFIS - Welcome to EFFIS," [Online]. Available: <http://effis.jrc.ec.europa.eu/>. [Acedido em 22 Julho 2019].

[14] "European Forest Fire Information System | Copernicus," [Online]. Available: <https://www.copernicus.eu/en/european-forest-fire-information-system>. [Acedido em 23 Julho 2019].

[15] "EFFIS - Technical Background," [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/>. [Acedido em 23 Julho 2019].

[16] "EFFIS - Fire Danger Forecast," [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/fire-danger-forecast/>. [Acedido em 25 Julho 2019].

[17] "EFFIS - Active Fire Detection," [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/active-fire-detection/>. [Acedido em 25 Julho 2019].

[18] "EFFIS - Rapid Damage Assessment," [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/rapid-damage-assessment/>. [Acedido em 25 Julho 2019].

[19] "EFFIS - Fire Damage Assesment," [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/fire-damage-assesment/>. [Acedido em 25 Julho 2019].

[20] "EFFIS - European Fire Database," [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/european-fire-database/>. [Acedido em 25 Julho 2019].

“EFFIS - Season Forecast,” [Online]. Available:
[21] <http://effis.jrc.ec.europa.eu/about-effis/technical-background/seasonal-forecast-explained/>. [Acedido em 25 Julho 2019].

“EFFIS - Monthly forecast,” [Online]. Available:
[22] <http://effis.jrc.ec.europa.eu/about-effis/technical-background/monthly-forecast-explained/>. [Acedido em 25 Julho 2019].

“EFFIS- Fuels,” [Online]. Available: <http://effis.jrc.ec.europa.eu/about-effis/technical-background/fuels/>. [Acedido em 25 Julho 2019].

“EFFIS- Applications,” [Online]. Available:
[24] <http://effis.jrc.ec.europa.eu/applications/>. [Acedido em 23 Julho 2019].

“Fogos.pt - Sobre,” [Online]. Available: <https://fogos.pt/sobre>. [Acedido em 07 Agosto 2019].

D. Kuhlman, em *A Python Book: Beginning Python, Advanced Python, and Python Exercises*, 2012, pp. 10-15.
[26]

“What is Python? Executive Summary | Python.org,” [Online].
[27] Available: <https://www.python.org/doc/essays/blurb/>. [Acedido em 30 Agosto 2019].

“Dynamic programming language - MDN Web Docs Glossary: Definitions of Web-related terms | MDN,” [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/Dynamic_programming_language. [Acedido em 30 Agosto 2019].

B. Venners, “The Making of Python,” artima, 13 Janeiro 2003. [Online].
[29] Available: <https://www.artima.com/intv/pythonP.html>. [Acedido em 30 Agosto 2019].

“General Python FAQ — Python 3.7.4 documentation,” [Online].
[30] Available: <https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place>. [Acedido em 30 Agosto 2019].

“The History of Python: A Brief Timeline of Python,” [Online]. Available:
[31] <https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html>. [Acedido em 30 Agosto 2019].

C. Fairchild, “Guido van Rossum Stepping Down from Role as Python's Benevolent Dictator For Life,” *Linuc Journal*, 12 Julho 2018.

A. Kuchling e M. Zadka, “What’s New in Python 2.0 — Python 3.7.4 documentation,” [Online]. Available: <https://docs.python.org/3/whatsnew/2.0.html>. [Acedido em 30 Agosto 2019].

“Python 3.0 Release | Python.org,” [Online]. Available: <https://www.python.org/download/releases/3.0/>. [Acedido em 30 Agosto 2019].

“FAQ: General | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/dev/faq/general/>. [Acedido em 12 Agosto 2019].

“The Web framework for perfectionists with deadlines | Django,” [Online]. Available: <https://www.djangoproject.com/>. [Acedido em 12 Agosto 2019].

“Django Software Foundation FAQ | Django,” [Online]. Available: <https://www.djangoproject.com/foundation/faq/>. [Acedido em 12 Agosto 2019].

“Design philosophies | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.0/misc/design-philosophies/>. [Acedido em 12 Agosto 2019].

“Top 10 sites built with Django Framework - DDI Development,” [Online]. Available: <http://ddi-dev.com/blog/programming/top-sites-built-django-framework/>. [Acedido em 12 Agosto 2019].

“Templates | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.0/topics/templates/#template-language-intro>. [Acedido em 13 Agosto 2019].

“GitHub - agilic/django-graphos: Django charting made *really* easy.,” [Online]. Available: <https://github.com/agilic/django-graphos>. [Acedido em 13 Agosto 2019].

[42] “GitHub - makinacorp/django-leaflet: Use Leaflet in your Django projects,” [Online]. Available: <https://github.com/makinacorp/django-leaflet>. [Acedido em 14 Agosto 2019].

[43] “Leaflet - a JavaScript library for interactive maps,” [Online]. Available: <https://leafletjs.com/>. [Acedido em 14 Agosto 2019].

[44] “GeoDjango Tutorial | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/tutorial/#introduction>. [Acedido em 14 Agosto 2019].

[45] “GeoDjango Installation | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/install/#overview>. [Acedido em 15 Agosto 2019].

[46] “Installing Geospatial libraries | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/install/geolibs/>. [Acedido em 15 Agosto 2019].

[47] “GDAL API | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/gdal/>. [Acedido em 15 Agosto 2019].

[48] “PROJ — PROJ 6.1.1 documentation,” [Online]. Available: <https://proj.org/>. [Acedido em 15 Agosto 2019].

[49] “PostGIS — Spatial and Geographic Objects for PostgreSQL,” [Online]. Available: <https://postgis.net/>. [Acedido em 15 Agosto 2019].

[50] “GeoDjango Model API | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/model-api/>. [Acedido em 15 Agosto 2019].

[51] “What is raster data?—Help | ArcGIS for Desktop,” [Online]. Available: <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm>. [Acedido em 15 Agosto 2019].

“GeoDjango Database API | Django documentation | Django,” [Online].
[52] Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/db-api/>.
[Acedido em 15 Agosto 2019].

“GeoDjango Forms API | Django documentation | Django,” [Online].
[53] Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/forms-api/>.
[Acedido em 15 Agosto 2019].

“GEOS API | Django documentation | Django,” [Online]. Available:
[54] <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/geos/>. [Acedido em 15
Agosto 2019].

“GeoDjango | Django documentation | Django,” [Online]. Available:
[55] <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/>. [Acedido em 15
Agosto 2019].

“LayerMapping data import utility | Django documentation | Django,”
[56] [Online]. Available:
<https://docs.djangoproject.com/en/2.2/ref/contrib/gis/layermapping/>.
[Acedido em 15 Agosto 2019].

“GeoDjango Management Commands | Django documentation |
[57] Django,” [Online]. Available:
<https://docs.djangoproject.com/en/2.2/ref/contrib/gis/commands/>. [Acedido
em 15 Agosto 2019].

“GeoDjango’s admin site | Django documentation | Django,” [Online].
[58] Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/admin/>.
[Acedido em 15 Agosto 2019].

“Geographic Sitemaps | Django documentation | Django,” [Online].
[59] Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/sitemaps/>.
[Acedido em 15 Agosto 2019].

“DGTerritório - Cartografia de Uso e Ocupação do Solo (COS, CLC e
[60] Copernicus),” [Online]. Available:
http://www.dgterritorio.pt/cartografia_e_geodesia/cartografia/cartografia_tematica/cartografia_de_uso_e_ocupacao_do_solo__cos_clc_e_copernicus_.
[Acedido em 16 Agosto 2019].

[61] “DGTerritório - Carta de Uso e Ocupação do Solo de Portugal Continental (COS),” [Online]. Available: http://www.dgterritorio.pt/dados_abertos/cos/. [Acedido em 16 Agosto 2019].

[62] M. Caetano, C. Igreja e F. Marcelino, “Carta de Uso e Ocupação do Solo de Portugal Continental para 2015,” em *Conferência Nacional de Cartografia e Geodesia*, Lisboa, 2015.

[63] “ICNF - Instituto da Conservação da Natureza e das Florestas,” [Online]. Available: <https://www.icnf.pt/quemsomos>. [Acedido em 28 Agosto 2019].

[64] Portugal, “Decreto-Lei n.º 135, de 29 de junho de 2012,” jun 2012. [Online]. Available: <https://dre.pt/legislacao-consolidada-/lc/107738445/201608120100/73456893/diploma/indice>. [Acedido em 28 Agosto 2019].

[65] “Django 2.2.2 release notes | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/releases/2.2.2/>. [Acedido em 12 Agosto 2019].

[66] “Databases | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/databases/>. [Acedido em 21 Agosto 2019].

[67] “Installing PostGIS | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/ref/contrib/gis/install/postgis/>. [Acedido em 24 Agosto 2019].

[68] “PostgreSQL + Python | Psycopg,” [Online]. Available: <http://initd.org/psycopg/>. [Acedido em 24 Agosto 2019].

[69] “GeoJSON,” [Online]. Available: <https://geojson.org/>. [Acedido em 26 Agosto 2019].

[70] “Serializing Django objects | Django documentation | Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/topics/serialization/>. [Acedido em 26 Agosto 2019].

FIRMS, “FIRMS - Fire Map,” [Online]. Available:
[71] https://firms.modaps.eosdis.nasa.gov/map/#z:7;c:-7.9,39.5;t:adv-grids;d:2019-07-15..2019-07-22;l:topo,firms_viirs. [Acedido em 19 Julho 2019].

“Discover Our Satellites | Copernicus,” [Online]. Available:
[72] <https://www.copernicus.eu/en/about-copernicus/infrastructure/discover-our-satellites>. [Acedido em 22 Julho 2019].

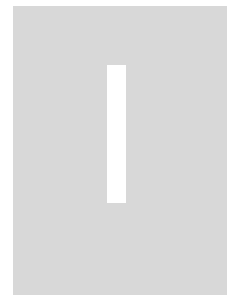
“Fogos.pt Início,” [Online]. Available:
[73] <https://fogos.pt/fogo/2019110160986>. [Acedido em 07 Julho 2019].

“Registo Nacional de Dados Geográficos - Direção-Geral do Território,” [Online]. Available:
[74] <https://snig.dgterritorio.gov.pt/rndg/srv/por/catalog.search#/map>. [Acedido em 16 Agosto 2019].

“EFFIS - Fire News,” [Online]. Available:
[75] <https://effis.jrc.ec.europa.eu/applications/fire-news/>. [Acedido em 25 Julho 2019].

“EFFIS - Current Situation,” [Online]. Available:
[76] https://effis.jrc.ec.europa.eu/static/effis_current_situation/public/index.html. [Acedido em 25 Julho 2019].

“Fogos.pt - Estatísticas,” [Online]. Available:
[77] <https://fogos.pt/estatisticas>. [Acedido em 7 Agosto 2019].



Anexo

Algoritmos de tratamento de dados

```
#####  
#Create Data frame with SubClass/year - Perc  
#####  
  
years=['2014','2015','2016','2017','2018']  
switcher={'2014':ardidoCOS2014,'2015':ardidoCOS2015,'2016':ardidoCOS2016,'2017':ardidoCOS2017,'2018':ardidoCOS2018}  
cos_mega=COS2015.COS2015_Le.unique()  
  
Areas = pd.DataFrame(index=years)  
for k in cos_mega:  
    Areas[k]=0.0  
for i in years:  
    #####  
    #create aux data frame to eliminate duplicates  
    ardidos_area=switcher[i].copy()  
    if i == '2014':  
        ardidos_area.drop_duplicates(subset ="COS_ID_fid", keep = "first", inplace = True)  
    else: ardidos_area.drop_duplicates(subset ="ID_COS_fid", keep = "first", inplace = True)  
    #####  
    for j in cos_mega:  
        if i == '2017':  
            total=ardidos_area[(ardidos_area.COS2015_Le==j)].AREA.sum()  
            burned=switcher[i][(switcher[i].COS2015_Le==j)].AREA_NOVA_CALC.sum()  
        else:  
            total=ardidos_area[(ardidos_area.COS2015_Le==j)].AREA.sum()  
            burned=switcher[i][(switcher[i].COS2015_Le==j)].area_calc_intersect.sum()  
        if (total!=0):  
            Areas[j][i]=burned/total*100  
        else:  
            Areas[j][i]=0  
Areas=Areas.sort_index()  
Areas
```

```
#####
#Create Data frame with Regions/sub - Perc
#####

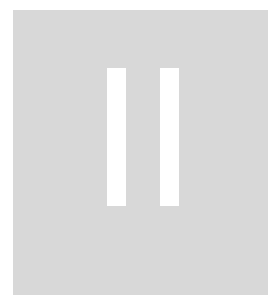
cos_codes=COS2015.layer.unique()
cos_sub=COS2015.COS2015_Le.unique()
#####
#create aux data frame to eliminate duplicates
ardidos_area=ardidoCOS2017.copy()
ardidos_area.drop_duplicates(subset ="ID_COS_fid", keep = "first", inplace = True)
#####
Areas = pd.DataFrame(index=cos_codes)

for k in cos_sub:
    Areas[k]=0.0
for i in cos_codes:
    for j in cos_sub:
        total=ardidos_area[(ardidos_area.layer==i) & (ardidos_area.COS2015_Le==j)].AREA.sum()
        burned=ardidoCOS2017[(ardidoCOS2017.layer==i) & (ardidoCOS2017.COS2015_Le==j)].AREA_NOVA_CALC.sum()
        if (total!=0):
            Areas[j][i]=burned/total*100
        else:
            Areas[j][i]=0
Areas=Areas.sort_index()
Areas
```

```
#####
#Create Data frame with MegaClass/year - Perc_country
#####

years=['2014', '2015', '2016', '2018']
switcher={'2014':ardidoCOS2014, '2015':ardidoCOS2015, '2016':ardidoCOS2016, '2018':ardidoCOS2018}
cos_mega=COS2015.Megaclasse.unique()

Areas = pd.DataFrame(index=years)
for k in cos_mega:
    Areas[k]=0.0
for i in years:
    #####
    #create aux data frame to eliminate duplicates
    ardidos_area=switcher[i].copy()
    if i == '2014':
        ardidos_area.drop_duplicates(subset ="COS_ID_fid", keep = "first", inplace = True)
    else: ardidos_area.drop_duplicates(subset ="ID_COS_fid", keep = "first", inplace = True)
    #####
    for j in cos_mega:
        total=COS2015[(COS2015.Megaclasse==j)].AREA.sum()
        burned=switcher[i][(switcher[i].Megaclasse==j)].area_calc_intersect.sum()
        if (total!=0):
            Areas[j][i]=burned/total*100
        else:
            Areas[j][i]=0
Areas=Areas.sort_index()
Areas
```



Anexo

As tabelas presentes na base de dados encontram-se disponíveis na seguinte Dropbox:

<https://www.dropbox.com/sh/oaegoagqju7vhl3x/AAAjj3gzUjxDJVEHPoBuNMSPa?dl=0>