



UNIVERSIDADE D
COIMBRA

Ricardo David da Silva Briceño

IMPLEMENTATION OF FIRELOC APPLICATION

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software orientada pelo Professor Doutor Filipe João Boavida de Mendonça Machado de Araújo e pelo Mestre Joaquim António Saraiva Patriarca apresentada à Faculdade de Ciências e Tecnologia do Departamento de Engenharia Informática.

Julho de 2022

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Implementation of the FireLoc Application

Ricardo David da Silva Briceño

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software orientada pelo Professor Doutor Filipe João Boavida de Mendonça Machado de Araújo e pelo Mestre Joaquim António Saraiva Patriarca apresentada à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Julho 2022



UNIVERSIDADE D
COIMBRA

Esta página foi deixada em branco intencionalmente..

Agradecimentos

Antes de mais, gostava de agradecer a todas as pessoas que tiveram a humildade e consideração em proporcionar, de alguma maneira, o seu apoio nesta experiência tão importante para mim.

Primeiramente, quero agradecer ao meu orientador, o Doutor Filipe Araújo, e ao meu coorientador, o Mestre Joaquim Patriarca, por toda a disponibilidade, pela assistência que revelaram, pela forma como me integraram na equipa e, acima de tudo, pelas orientações que me fizeram refletir e melhorar a nível pessoal e profissional. Agradeço à professora Cidália Fonte e a toda a equipa FireLoc por terem acreditado em mim e pela oportunidade de poder contribuir num projeto dedicado a uma causa incrível.

Agradeço à minha família e aos meus amigos por me terem incentivado a lutar pelos meus sonhos e objetivos. À família Farinha quero, igualmente, dar a minha gratidão por todo o carinho e ajuda numa fase diferente da minha vida e por me fazerem sentir “em casa”. Por último, queria agradecer à minha mãe, ao meu pai e ao meu irmão por todo o seu amor e suporte que me têm dado, incondicionalmente. A pessoa que me tornei é um reflexo de tudo o que têm feito. Todo o meu trabalho e esforço é dedicado a vocês, assim como os seus resultados.

Este trabalho decorreu no âmbito do projeto FireLoc (PCIF/MPG/0128/2017), financiado pela Fundação para a Ciência e a Tecnologia.

Esta página foi deixada em branco intencionalmente..

Abstract

Forest fires are one of the main threats to the biodiversity of non-fire-dependent ecosystems and to the population. This threat has been increasingly taken into account since its consequences compromise the safety of the population and the sustainability of nature, harming socio-economic and socio-cultural activities. Fireloc aims to provide a crowdsourced forest fire detection and prevention system that allows the contribution of the general population to assist in the management and prevention of fires. The system is served by a wide variety of sources that allows providing risk authorities with suitable tools for analysis and operation of action on fire occurrences. This is through two applications, a mobile application for data collection data (contributions) possible from users and a portal with two purposes, administrative and fire management and informational purposes for common users.

This dissertation describes the current state of the FireLoc system, allowing to identify, compare and highlight its virtues and recognize possible similarities with other systems. A specification of the system's functionalities was also carried out, namely the back-end for the management and processing of geospatial information through the development of API FireLoc and the creation of services with GeoServer to be consumed on the front-end. It should be noted that a representation of the FireLoc system's software architecture design is made with different architectural views to clarify the system's operation and identify the quality attributes that the system must guarantee. And, finally, all the modules developed were tested to verify the proper functioning and certify that all the defined objectives were reached.

Keywords

Forest Fires. Web Development. Back-end. API. Geographic Information Systems.

Esta página foi deixada em branco intencionalmente..

Resumo

Os incêndios florestais são uma das principais ameaças à biodiversidade de ecossistemas não dependentes do fogo assim como à população. Esta ameaça tem sido, cada vez mais, levada em consideração uma vez que as suas consequências têm comprometido a segurança da população e a sustentabilidade da natureza, prejudicando diversas atividades (sócio-económicas e sócio-culturais). O projeto Fireloc tem como objetivo providenciar um sistema de deteção e prevenção de incêndios florestais por crowdsourcing que permite a contribuição da população geral para auxiliar na gestão e prevenção de incêndios. O sistema é servido de uma grande variedade de fontes de dados que permite providenciar às autoridades as devidas ferramentas para análise de risco e planeamento de atuação sobre as ocorrências de incêndios. Isto é possível através de duas aplicações, uma aplicação móvel para a recolha de dados (contribuições) dos utilizadores e um portal com duas finalidades, fins administrativos e de gestão de incêndios e fins informativos para utilizadores comuns.

A presente dissertação descreve o estado atual do sistema FireLoc, permitindo identificar, comparar e realçar as suas virtudes e reconhecer eventuais semelhanças com outros sistemas. Foi também realizada uma especificação das funcionalidades do sistema, nomeadamente o back-end para gestão e processamento de informação geospacial através do desenvolvimento da API FireLoc e criação de serviços com o GeoServer para serem consumidos no front-end. De salientar que é feita uma representação do desenho da arquitetura de software do sistema FireLoc com diferentes vistas arquiteturais para clarificar o funcionamento do sistema e identificação de atributos de qualidade que o sistema deverá garantir. E, finalmente, foram testados todos os módulos desenvolvidos para verificar o bom funcionamento e certificar que todos os objetivos definidos foram atingidos.

Palavras-Chave

Incêndios Florestais. Desenvolvimento Web. Back-end. API. Sistemas de Informação Geográfica.

Esta página foi deixada em branco intencionalmente..

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação	2
1.3	Objetivos Gerais	3
1.3.1	Objetivos Específicos	3
1.4	Estrutura do Documento	4
2	Revisão Sistemática da Literatura	6
2.1	Deteção de Incêndios	6
2.2	Sistema FireLoc	7
2.3	Aplicações existentes	11
2.3.1	Aplicação Fogos	11
2.3.2	Sistema GesFoGO	12
2.3.3	Sistema Bee2Fire	13
2.3.4	Aplicação Fire RoS Calculator - ADAI	14
2.3.5	Sistema IMfire - ADAI	15
2.4	Comparação entre sistemas	16
2.4.1	Sumário	19
2.5	Tecnologias	20
2.5.1	Tecnologias em Web Mapping	20
2.5.2	Tecnologias em Front-end	31
2.5.3	Tecnologias em Back-end	33
2.5.4	Base de Dados	34
2.5.5	Tecnologias de Virtualização	35
2.5.6	Sumário	36
3	Especificação de Drivers Arquiteturais	37
3.1	Requisitos Funcionais - Portal FireLoc	37
3.1.1	UC - 1: Autenticar	38
3.1.2	UC - 2: Gerir Utilizadores	39
3.1.3	UC - 3: Gerir Grupos	40
3.1.4	UC - 4: Gerir Contribuições	41
3.1.5	UC - 5: Gerir Eventos	42
3.1.6	UC - 6: Gerir Camadas	44
3.1.7	UC - 7: Gerir Gráficos	46
3.1.8	UC - 8: Gerir Resultados do Processo de Localização de Eventos	47
3.1.9	UC - 9: Gerir Dados Raster	49
3.1.10	UC - 10: Gerir Informação Meteorológica	50
3.1.11	UC - 11: Gerir Resultados do Processo da Classificação de Fotografias	52
3.2	Critérios de Aceitação	53
3.3	Atributos de Qualidade	54

3.3.1	Interoperabilidade	54
3.3.2	Performance	55
3.3.3	Segurança	56
3.3.4	Disponibilidade	57
3.3.5	Priorização de Atributos de Qualidade	58
4	Arquitetura do Sistema	60
4.1	Modelo C4 - Context, Container, Components, Code	60
4.1.1	Diagrama de Contexto do sistema	60
4.1.2	Diagrama de Container	62
4.1.3	Diagrama de Componentes	63
4.2	Modelo C&C - Componentes e Conectores	64
4.2.1	3 Tier - Shared Data View	65
4.3	Comparação entre Modelos	66
4.4	Validação da Arquitetura - Traceability	67
5	Implementação	69
5.1	Desenvolvimento da API	69
5.1.1	Aplicações Externas	69
5.1.2	Aplicações Internas	70
5.1.3	Boas Práticas utilizadas	71
5.1.4	Authapi	72
5.1.5	Contrib	74
5.1.6	Events	76
5.1.7	Firedetect	77
5.1.8	GeoRef	78
5.1.9	GeoRst	80
5.1.10	GeoVis	82
5.1.11	Meteo	83
5.1.12	SDI	84
5.1.13	Módulo de Recolha e Harmonização dos dados	86
5.2	Desenvolvimento com o GeoServer	86
5.2.1	Criação de Serviços - Mapas	87
5.3	Desenvolvimento com Docker	87
5.3.1	Cron Job Services	88
5.3.2	Aplicações de Catálogo	89
5.4	Sistema FireLoc em Produção	90
5.4.1	Deployment do Sistema	90
5.4.2	Desafios Associados	91
6	Validação do Sistema	92
6.1	Testes da API	92
6.1.1	Fase de Verificação	92
6.1.2	Fase de Validação	93
6.1.3	Testes Unitários	93
6.1.4	Testagem com Jupyter Lab - Notebooks	93
6.1.5	Testagem com Postman - Swagger	94
6.2	Testes de Integração	98
6.3	Testes de Atributos de Qualidade	98
6.3.1	Testes de Segurança no Sistema	99
6.3.2	Testes da Disponibilidade no Sistema	100
6.3.3	Testes da Interoperabilidade no Sistema	101

6.3.4	Testes de Performance no Sistema	101
7	Conclusão	105
7.1	Análise do Planeamento	105
7.2	Reflexão Final	107
7.3	Trabalho Futuro	107

Esta página foi deixada em branco intencionalmente..

Acrónimos

- ACID** Atomicidade, Consistência, Isolamento e Durabilidade. 57, 99
- ADAI** Associação para o Desenvolvimento da Aerodinâmica Industrial. ix, 14, 15, 17, 18
- API** Application Programming Interface. x, 3, 4, 10, 12, 17–19, 23, 25, 28–30, 36, 38, 44, 50–52, 55–57, 59, 61–65, 68, 69, 71–73, 83–87, 90, 91, 93–95, 97, 99–102, 104, 107, 108
- AWS** Amazon Web Services. 35, 36, 100
- CAOP** Carta Administrativa Oficial de Portugal. 78
- CEIF** Centro de Estudos sobre Incêndios Florestais. 14
- CMS** Content management system. 29, 30
- COS** Cartografia de Uso e Ocupação do Solo. 23, 80
- CQL** Common Query Language. 103
- CRS** Coordinate reference system. 8, 24, 75
- CRUD** Create, Read, Update and Delete. 21, 39, 41, 47, 49, 52, 67, 72, 85
- DGT** Direção Geral do Território. 23, 61
- DOM** Document Object Model. 103
- Eelab** Enterprise Engineering Laboratory. 12
- EPSG** European Petroleum Survey Group. 24, 75, 77
- FCTUC** Faculdade de Ciências e Tecnologia da Universidade de Coimbra. 14
- GDAL** Geospatial Data Abstraction Library. 25, 33, 75, 76, 81, 82
- GesFoGO** Gestão de incêndios Florestais mediante Georreferenciação em Observadores móveis. 12, 19
- GIS** Geographic Information System. 3, 7, 11, 21–23, 25, 27, 33, 34, 40, 59, 75, 101, 107
- GNR** Guarda Nacional Republicana. 2
- GPKG** GeoPackage. 80–82, 85
- GPS** Global Positioning System. 17, 31
- GRASS** Geographic Resources Analysis Support System. 25, 33, 75, 88
- HTTP** Hypertext Transfer Protocol. 21, 27, 28, 54, 71, 73, 74, 90, 91, 95, 97, 99

HTTPS HyperText Transfer Protocol Secure. 91

ICNF Instituto da Conservação da Natureza e das Florestas. 1, 2, 12, 13, 40, 77, 86

IPMA Instituto Português do Mar e da Atmosfera. 50, 83, 84, 86

ITI Interactive Technologies Institute. 12

JSON JavaScript Object Notation. 71, 84, 86, 95

MTV Model-Template-View. 33, 60

MVC Model-View-Controller. 33

NPM Node Package Manager. 34

NUT Nomenclatura das Unidades Territoriais para Fins Estatísticos. xvi, 78, 79

OGC Open Geospatial Consortium. 8, 20, 21, 25, 27–31, 54, 55, 59, 62, 76, 80, 101

ORM Object-relational mapping. 33, 72

OS Operating System. 9, 26, 28, 107

OSM Open Street Map. 17, 61

OWASP Open Web Application Security Project. 99

PCP Proteção Cível Portuguesa. 2, 7, 9, 11–13, 40, 55, 62

REST Representational State Transfer. 29, 62, 65, 76, 85–87, 102

RF Requisito Funcional. 38, 39, 53, 54, 92

SDI Spatial Data Infrastructure. 10, 20, 21, 27, 29, 36, 63, 65, 70, 82, 84, 87

SMTP Simple Mail Transfer Protocol. 73

SNB Serviço Nacional de Bombeiros. 2

SQL Structured Query Language. 81, 91

SSL Secure Socket Layer. 72, 91

UC Use Case. xvi, xix, 38–40, 42, 44, 48, 50–54, 67, 68, 74, 116

UI User Interface. xvi, 9, 12, 14, 16–18, 31, 58, 89, 106

ULPGC Universidade de Las Palmas de Gran Canaria. 12

WCS Web Coverage Service. 21–23

WFS Web Feature Service. xvi, 21, 22, 26, 27, 86, 101, 103

WKT Well-known text. xvi, 76

WMS Web Map Service. xvi, 21–23, 25, 27, 28, 86, 101

XML Extensible Markup Language. 85, 86

Esta página foi deixada em branco intencionalmente..

Lista de Figuras

1.1	Imagem retirada de FIRELOC - Localize o Fogo [24]	2
2.1	Aplicação Móvel - Verificação da Orientação ao Norte	8
2.2	Prototipagem da User Interface (UI) Portal - ZIP Design: Creativity Agency	9
2.3	Arquitetura do Sistema FireLoc	10
2.4	O mapa de incêndios ativos no território português é aqui visto em conjunto com a informação da temperatura atmosférica	11
2.5	Interface Gráfica do Utilizador GesFoGO - Imagem retirada de [75]	12
2.6	Processo de deteção GesFoGO - Imagem retirada de [75]	13
2.7	Imagem retirada de Bee2Fire -Inteligência artificial [49]	14
2.8	Imagem retirada de: A Tool to Measure the Rate of Spread of a Propagating Wildfire in a Laboratory Setting [3]	15
2.9	Imagem retirada de IMfire - Gestão de incêndios [61]	15
2.10	Imagem retirada de 2nd Workshop – Scientific Research and Technological Development - Forest Fire Prevention and Fire Fighting [76]	16
2.11	Imagem retirada de [24] - Posicionamento	18
2.12	Comparação entre Web Map Service (WMS) e Web Feature Service (WFS) [Adaptado de [69]]	21
2.13	Resultados dos serviços WMS e WFS	22
2.14	Raster em Modelo Digital de Elevação [6]	23
2.15	Dados vetoriais - Features	24
2.16	Interação entre o Gerador de Serviços e Leaflet	27
3.1	Níveis de abstração: Use Case (UC)	38
4.1	Diagrama de Contexto	61
4.2	Diagrama de Container	62
4.3	Diagrama de Componentes	64
4.4	Notação 3 Tier - Shared Data View	65
4.5	Diagrama C&C - 3-Tier com Shared Data	66
5.1	Diferentes tipos de projeções [67]	75
5.2	Representação em texto de geometrias vetoriais 2D - Well-known text (WKT) [87]	76
5.3	Modelos Nomenclatura das Unidades Territoriais para Fins Estatísticos (NUT)s - Imagem adaptada de [74]	79
5.4	Grelha de Referência Geográfica para Portugal Continental	79
5.5	Log do Container para execução de Cron Jobs.	89
5.6	Renderização de Estações Meteorológicas no GeoNode	90
6.1	Ciclo de Testes	93
6.2	Resultados da Automatização de Testes - Exemplo	95

6.3	Resultados para o método GET (listagem) - Boxplot	96
6.4	Medição da Disponibilidade - Monitorização da API FireLoc	100
6.5	Esquema de processamento com GeoWebCache	102
6.6	Marker Clustering - Exemplo	103
6.7	Bounding Box - Figura adaptada de [45]	104
7.1	Diagrama de Gantt - Plano idealizado	105
7.2	Diagrama de Gantt - Plano efetivo	106
3	Diagrama UC - 1: Autenticação	116
4	Diagrama UC - 2: Gerir utilizadores	116
5	Diagrama UC - 3: Gerir grupos	116
6	Diagrama UC - 4: Gerir Contribuições	117
7	Diagrama UC - 5: Gerir Eventos	118
8	Diagrama UC - 6: Gerir Camadas Geoespaciais	119
9	Diagrama UC - 7: Gerir gráficos	120
10	Diagrama UC - 8: Gerir Resultados do Processo de Localização de Eventos	121
11	Diagrama UC - 9: Gerir Dados Raster	122
12	Diagrama UC - 10: Gerir Gerir Informação Meteorológica	123
13	Diagrama UC - 11: Gerir Resultados do Processo da Classificação de Foto- grafias	124
14	Resultados para o método POST (Adicionar) - Boxplot	184
15	Resultados para o método POST (Adicionar Ficheiros Raster) - Boxplot . .	184

Esta página foi deixada em branco intencionalmente..

Lista de Tabelas

2.1	Comparação entre os serviços de renderização de mapas	26
2.2	Comparação entre os servidores de que geram os serviços de dados geoespaciais	28
2.3	Comparação entre as aplicações de catálogo - Baseado na informação retirada de [26] [29]	30
2.4	Comparação entre aplicações: Nativa, Web e Híbrida [72]	32
2.5	Comparação entre frameworks de server-side	34
2.6	Tecnologias escolhidas por vertente tecnológica	36
3.1	Cenário de Atributo de Qualidade - Interoperabilidade	55
3.2	Cenário de Atributo de Qualidade - Performance (Renderização)	55
3.3	Cenário de Atributo de Qualidade - Performance (Tempo de Resposta)	56
3.4	Cenário de Atributo de Qualidade - Segurança (Confidencialidade)	56
3.5	Cenário de Atributo de Qualidade - Segurança (Integridade dos dados)	57
3.6	Cenário de Atributo de Qualidade - Segurança (Integridade do sistema)	57
3.7	Cenário de Atributo de Qualidade - Disponibilidade	58
3.8	Priorização de Brosseau - Atributos de Qualidade	58
4.1	Traceability - Relação entre Funcionalidades e UCs	68
5.1	Aplicações externas - Django	70
5.2	Aplicações internas - Sistema FireLoc	70
5.3	Códigos de Sucesso	71
5.4	Sintaxe do Cron Tab	88
1	Use Case 1.1 - Login	125
2	Use Case 1.2 - Registrar	126
3	Use Case 1.3 - Recuperar password	127
4	Use Case 2.1 - Adicionar um novo utilizador	128
5	Use Case 2.2 - Atribuir grupo a um utilizador	129
6	Use Case 2.3 - Atualizar utilizador	130
7	Use Case 2.4 - Excluir utilizador	131
8	Use Case 2.5 - Listar utilizadores	132
9	Use Case 2.6 - Obter um utilizador específico	133
10	Use Case 3.1 -Listar os grupos de utilizador	134
11	Use Case 3.2 - Adicionar um novo grupo	135
12	Use Case 3.3 - Atualizar grupo	136
13	Use Case 3.4 - Excluir grupo	137
14	Use Case 4.1 - Lista contribuições voluntárias	138
15	Use Case 4.2 - Representar contribuições voluntárias	139
16	Use Case 4.3 - Obter contribuição específica	140
17	Use Case 4.4 - Excluir contribuições	141
18	Use Case 4.5 - Adicionar contribuições	142

19	Use Case 5.1 - Gerir Eventos Reais (Observados à priori)	143
20	Use Case 5.3 - Gerir Anos)	144
21	Use Case 5.4 - Gerir Áreas Ardidadas	145
22	Use Case 6.1 - Adicionar uma nova camada geospacial	146
23	Use Case 6.2 - Atualizar uma camada geospacial	147
24	Use Case 6.3 - Excluir uma camada geospacial	148
25	Use Case 6.4 - Listar camadas geospaciais	149
26	Use Case 6.5 - Obter uma camada geoespacial específica	150
27	Use Case 6.6 - Atribuir permissões de visualização	151
28	Use Case 6.7 - Gerir estilos	152
29	Use Case 6.8 - Gerir Legendas	153
30	Use Case 7.1 - Adicionar um novo gráfico geospacial	154
31	Use Case 7.2 - Adicionar dados a um gráfico geospacial	155
32	Use Case 7.3 - Atualizar gráfico geospacial	156
33	Use Case 7.4 - Excluir gráficos geospaciais	157
34	Use Case 7.5 - Atribuir requisitos de permissões de visualização	158
35	Use Case 7.6 - Gerir Dados	159
36	Use Case 8.1 - Gerir Atributos da Avaliação	160
37	Use Case 8.2 - Gerir Resultados da Avaliação	161
38	Use Case 8.3 - Gerir Abordagens da Avaliação	162
39	Use Case 8.4 - Gerir Procedimentos	163
40	Use Case 9.1 - Gerir Tipos Raster	164
41	Use Case 9.2 - Gerir Datasets Raster	165
42	Use Case 9.3 - Gerir Camadas Raster	166
43	Use Case 9.4 - Adicionar Ficheiros Raster	167
44	Use Case 10.1 - Gerir Previsões Meteorológicas	168
45	Use Case 10.2 - Gerir Observações Meteorológicas	169
46	Use Case 10.3 - Gerir Fontes Meteorológicas	170
47	Use Case 10.4 - Gerir estações Meteorológicas	171
48	Use Case 10.5 - Gerir Variáveis Meteorológicas	172
49	Use Case 11.1 - Gerir Atributos da Classificação	173
50	Use Case 11.2 - Gerir Resultados da Classificação	174
51	Use Case 11.3 - Gerir Procedimentos	175
52	Endpoints do Sistema de Autenticação e Autorização	176
53	Endpoints da Gestão de Utilizadores	176
54	Endpoints da Gestão de Contribuições	176
55	Endpoints da Gestão de Eventos Reais	177
56	Endpoints Gestão de Anos dos Eventos	177
57	Endpoints da Gestão de Áreas Ardidadas	177
58	Endpoints Gestão de Atributos Avaliação de Identificação de Incêndios . . .	177
59	Endpoints Gestão de Abordagens de Avaliação de Identificação de Incêndios	178
60	Endpoints Gestão de Resultados de Avaliação de Identificação de Incêndios	178
61	Endpoints Gestão de Procedimentos de Avaliação de Identificação de Incêndios	178
62	Endpoints Gestão de Atributos da Classificação de Fotografias de Incêndios	178
63	Endpoints Gestão de Resultados da Classificação de Fotografias de Incêndios	179
64	Endpoints Gestão de Procedimentos da Classificação de Fotografias de In- cêndios	179
65	Endpoints Gestão da Grelha de Referência	179
66	Endpoints Gestão de Tipos Raster	179
67	Endpoints Gestão de Datasets Raster	179
68	Endpoints Gestão de Tipos Raster	180

69	Endpoints Gestão de Gráficos	180
70	Endpoints Gestão de Dados de Gráficos	180
71	Endpoints Gestão de Camadas	180
72	Endpoints Gestão de Permissões das Camadas	180
73	Endpoints Gestão de Legendas	181
74	Endpoints Gestão de Observações Meteorológicas	181
75	Endpoints Gestão de Previsões Meteorológicas	181
76	Endpoints Gestão de Fontes Meteorológicas	181
77	Endpoints Gestão de Estações Meteorológicas	182
78	Endpoints Gestão de Variáveis Meteorológicas	182
79	Endpoints Gestão de Workspaces do GeoServer	182
80	Endpoints Gestão de Stores do GeoServer	182
81	Endpoints Gestão de Camadas do GeoServer	182
82	Endpoints Gestão de Serviços do GeoServer	183

Blocos de Código

5.1	Uso de nomes em endpoints - Exemplo	71
5.2	PostGIS Query - Comparar se a distância entre as coordenadas e a geometria da célula é igual a 0 (0.000001)	81
5.3	Cron tab do sistema FireLoc	88
6.1	Test-Suite - Exemplo	94

Esta página foi deixada em branco intencionalmente..

Capítulo 1

Introdução

A presente dissertação está enquadrada na unidade curricular “Dissertação/Estágio” em Engenharia de Software do Mestrado em Engenharia Informática com ramo de especialização em Engenharia de Software, no Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, no ano letivo 2021/2022.

No desenvolvimento desta dissertação vai relatar-se o conhecimento adquirido na realização das tarefas definidas do plano de trabalhos estabelecido. Irá salientar-se todo o processo de trabalho, assim como a participação e envolvimento que foi surgindo, terminando com observações para trabalho futuro relativamente à área de estudo.

1.1 Contexto

Os incêndios florestais são considerados catástrofes naturais pelo facto de se desenvolverem na natureza, pela possibilidade de ocorrência e de que as características de propagação dependem, fortemente, de fatores naturais. Por outro lado, apesar de que as condições físicas sejam mais ou menos favoráveis a um incêndio, há que destacar que, na grande maioria dos casos, não são causas naturais as que provocam o fogo, mas sim a ação humana, quer seja acidentalmente ou não. Segundo dados do Instituto da Conservação da Natureza e das Florestas (ICNF), cerca de 70% dos casos em 2021 foram causados pelos humanos, nos quais as motivações são variadas, sendo as mais comuns: a queima não autorizada, ilegal e descontrolada de superfícies agrícolas, tais como uso negligente do fogo e o incendiarismo [18].

Como consequência, existe uma lista extensa de prejuízos ambientais, materiais e sociais, desde a destruição de campos agrícolas e da biodiversidade até a exposição de habitações e necessidade de evacuação da população local. Desta forma, todas as atividades relacionadas com a gestão e prevenção de incêndios florestais têm como principal foco a proteção da biodiversidade e da população local e, por essa razão, devem ser consideradas as diversas causas e agir com a maior brevidade possível de modo a reduzir os custos que as suas consequências possam vir a causar.

A prática de análise, gestão e prevenção de incêndios florestais é bastante peculiar, não tanto pelo facto de se tratar de incêndios florestais, mas sim por todos os desafios que estas práticas envolvem. Os dados disponibilizados no que concerne a incêndios florestais estão, na grande maioria nos casos, num estado bruto, isto é, sem tratamento prévio, o que obriga a que seja feito um processamento individual e adaptado ao contexto em que

será utilizado, para uma análise mais apropriada. A gestão da informação obtida tem de ter em conta uma série de condições que nem sempre podemos controlar e que acabam por influenciar a qualidade dos dados e o processo de tomadas de decisões.

1.2 Motivação

A logística florestal é uma área que está a crescer, acentuadamente, não apenas em Portugal, mas em todo o mundo. Para dar resposta ao problema deve-se planificar o espaço florestal, mediante projetos florestais. Estes projetos consideram, inicialmente, os locais mais propícios à ocorrência de incêndios e, posteriormente, procuram evitá-los, ou seja, há uma aposta muito forte na prevenção [9].

Neste contexto, o projeto Fireloc, financiado pela Fundação para a Ciência e a Tecnologia, apresenta um sistema que permite a gestão e armazenamento de dados georreferenciados para a prevenção e gestão de incêndios florestais que, por sua vez, irão auxiliar as entidades competentes e especializadas da área como o ICNF, Proteção Cívil Portuguesa (PCP), Serviço Nacional de Bombeiros (SNB) e Guarda Nacional Republicana (GNR).

O projeto florestal do Fireloc tem como objetivo integrar módulos de recolha, processamento e saída de dados com protocolos definidos, com consenso, para disponibilizar conteúdo e serviços geomáticos. Na figura 1.1 podemos observar, de forma generalizada, o esquema de comunicação entre os módulos do projeto FireLoc.

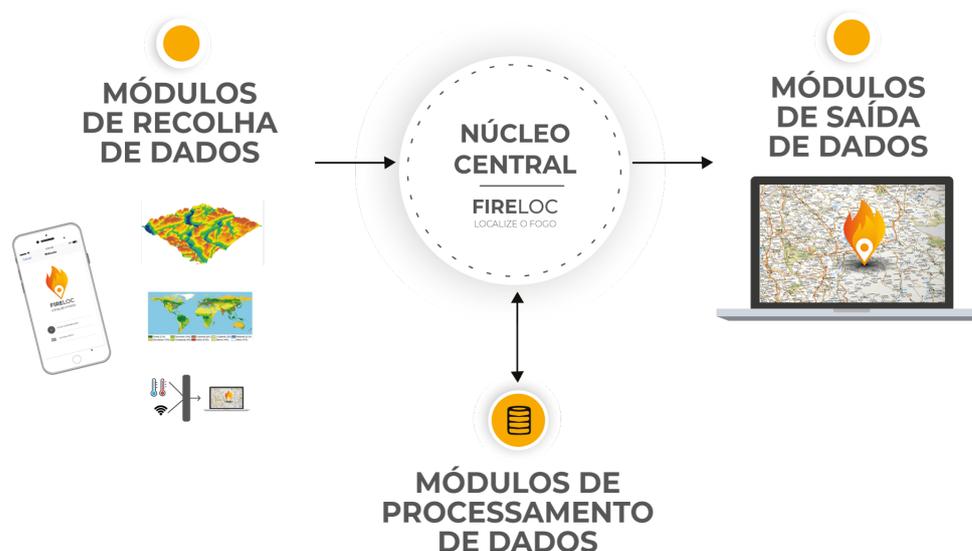


Figura 1.1: Imagem retirada de FIRELOC - Localize o Fogo [24]

O sistema Fireloc é inovador no que toca à recolha de dados. A recolha é feita a partir de uma aplicação móvel, sendo alguns dados automaticamente disponibilizados pelo dispositivo, como as coordenadas da sua localização (local de observação), a orientação do dispositivo e ainda a distância aproximada do evento que observa (indicada pelo observador) [12]. Para além disso, o sistema recolhe dados úteis de fontes externas para complementar os serviços disponíveis, atuando como intermediários para disponibilizar conteúdo aos utilizadores finais. Desta forma, o desenvolvimento deste sistema irá dispo-

nibilizar à população portuguesa uma ferramenta útil para qualquer cidadão que deseje denunciar a ocorrência de um incêndio florestal e, também, auxiliar entidades profissionais responsáveis para o combate e prevenção de incêndios florestais. Para tal, o projeto Fire-Loc oferece uma aplicação web (Portal) na qual se consegue monitorizar e fazer a gestão de informação georreferenciada em Portugal usando várias fontes de dados, nomeadamente redes de monitorização, mapas de uso, a recolha de dados feita pela aplicação móvel, entre outros. A aplicação web terá duas vertentes funcionais e destinadas para diferentes tipos de utilizador. Uma será destinada para entidades profissionais competentes e a outra para utilizadores comuns, onde estes últimos poderão ter acesso a funcionalidades de representação de incêndios florestais em Portugal, alguma informação estatística sobre os mesmos e outro tipo de informação adicional útil relacionada com os incêndios.

1.3 Objetivos Gerais

Esta dissertação tem como principal objetivo desenvolver diversas atividades no Módulo de Processamento de Dados. Este módulo está diretamente associado à vertente de Back-end do sistema, isto é, toda a parte que envolva o servidor, o que implica que também sejam realizadas atividades no módulo de Recolha de Dados, pois ambos estão relacionados.

1.3.1 Objetivos Específicos

O sistema Fireloc trata-se de um projeto que se encontra num estado avançado em termos de desenvolvimento, já estão implementados alguns recursos que poderão auxiliar na concretização dos objetivos específicos.

É importante destacar que a aplicação móvel já se encontra desenvolvida para dispositivos móveis Android e não será, naturalmente, foco de estudo. Desta forma, o cerne do trabalho será o desenvolvimento e melhoria dos módulos e componentes do servidor do sistema para que se consigam processar e gerir os dados internos das aplicações móvel e web e de outras aplicações externas. Sendo este o foco de trabalho, o aluno é responsável, principalmente, pelo desenvolvimento da Application Programming Interface (API) e processamento de informação geospacial com tecnologias Geographic Information System (GIS) para a criação de sistemas de gestão de serviços e renderização desta informação em mapas. As atividades com o servidor implicam que sejam feitas tarefas complementares, mas que não deixam de ser imprescindíveis, como é o caso da virtualização de componentes do servidor em containers, resultando na utilização de tecnologias de virtualização complexas como Docker. Em resumo, podemos destacar as seguintes tarefas:

- Preparação para as tecnologias GIS como o Leaflet para representação de dados em mapas através da web e o GeoServer para o desenvolvimento de soluções de web mapping mediante a criação de serviços, integrando diversos repositórios de dados geográficos e ainda compreensão da extensão do PostgreSQL (PostGIS) para armazenamento e manipulação de dados geográficos/espaciais, entre outros dados;
- Identificação de melhorias na API e aplicação das mesmas conforme necessidade;
- Contribuição no projeto com o desenvolvimento de módulos e/ou componentes;
- Testes de implementação e documentação dos módulos e/ou componentes desenvolvidos;

- Criação de serviços para consumo de mapas no front-end, nomeadamente com o GeoServer;
- Testes de renderização de dados geospaciais;
- Identificação de técnicas e estratégias de renderização de informação geoespacial no lado do GeoServer para melhorar a performance de renderização de dados geospaciais;
- Desenvolvimento de serviços cronológicos com finalidades variadas (obtenção e atualização de informação de forma periódica);
- Adaptar aplicações externas e containerização das mesmas para uso singular e exclusivo no sistema FireLoc;
- Configuração e execução das aplicações containerizadas em modo produção para acesso público.

1.4 Estrutura do Documento

A presente dissertação está estruturada em sete capítulos que por sua vez estão divididos em sub-seções.

O Capítulo 2 — descreve o estado atual de investigação científica relativamente a áreas de prevenção e gestão de incêndios florestais mediante sistemas informáticos e a identificação de pontos importantes neste campo que possam servir de exemplo ou meio de comparação com o sistema FireLoc. Também inclui uma revisão e descrição detalhada das tecnologias utilizadas no projeto FireLoc.

O Capítulo 3 — lista e descreve os requisitos funcionais do sistema identificados, de modo a ter um registo das necessidades atuais do sistema FireLoc. Também inclui a especificação de atributos de qualidade que o sistema deverá garantir, assim como a definição de critérios de aceitação sob as funcionalidades desenvolvidas.

O Capítulo 4 — clarifica a arquitetura atual do sistema FireLoc, no qual se apresentam diferentes tipos de informação mediante modelos e vistas arquiteturais, seguida de uma descrição detalhada das mesmas.

O Capítulo 5 — especifica todo o processo de implementação e desenvolvimento de cada uma das tarefas realizadas ao longo do estágio, nomeadamente com a API, o GeoServer e tarefas relacionadas com Docker.

O Capítulo 6 — representa todos os testes realizados na fase de implementação para garantir o funcionamento dos módulos desenvolvidos, assim como a qualidade do sistema mediante os atributos de qualidade identificados no Capítulo 3. Será destacado o processo de automatização para a realização dos testes e os resultados de cada um.

O Capítulo 7 — relata o esforço realizado no projeto, assim como os resultados alcançados com o plano de trabalho, incluindo uma revisão do mesmo, uma análise crítica dos objetivos alcançados, seguida de uma série de observações e sugestões que possam contribuir para trabalho futuro.

Esta página foi deixada em branco intencionalmente..

Capítulo 2

Revisão Sistemática da Literatura

Alguns investigadores científicos do Instituto Superior de Agronomia de Lisboa realizaram uma investigação [47] em relação aos Incêndios Florestais em Portugal, onde foi abordada a evolução da floresta ao longo do tempo. São identificadas algumas evidências que permitem compreender o porquê de Portugal ser tão prejudicado pelos incêndios florestais.

Portugal é um dos países que integra o Mediterrâneo e, devido à sua localização geográfica, é fortemente afetado por temperaturas altas em algumas épocas do ano. Para além disso, tem-se verificado uma grande diminuição das atividades agrícolas, que manteria o terreno livre de vegetação junto a povoações [47], e não podemos esquecer o facto das ações humanas terem um papel importante e influente nos incêndios florestais. Juntando estes três fatores é expectável que os incêndios em Portugal sejam algo muito comum e que facilmente tenham efeitos negativos [47].

O desenvolvimento de algumas áreas científicas, como a Palinologia Forense e a Antracologia, relatam que as intervenções humanas modificaram a floresta ao longo dos anos pela atividade agrícola (não tão significativo), pela urbanização e não só. A Palinologia é uma área que se ocupa do estudo do pólen e outros palinóforos. Estas micro-estruturas são muitas vezes estudadas para agrupar um conjunto de evidências que possam relatar acontecimentos relacionados com a natureza criminal [89]. Com estas evidências, é plausível provar a existência ou inexistência de relação entre vítimas, suspeitos, testemunhas, objetos e/ ou locais, sendo uma grande ajuda para suspeitas de incêndios [89]. Por outro lado, a Antracologia foca-se no estudo de resíduos/destruções de material lenhoso carbonizado. Estes resíduos ardidos são fortes testemunhos de incêndios, naturais ou de origem entrópica, sendo também imprescindíveis para a identificação das causas de incêndios florestais [88].

A exploração das florestas compromete e afeta, seriamente, a gestão e, principalmente, a prevenção de incêndios, uma vez que, deve ser realizado um estudo do território com frequência e re-avaliar o risco de incêndio de todas as zonas do país [47].

2.1 Deteção de Incêndios

Nos últimos anos, tem havido uma aposta positiva na prevenção de incêndios em Portugal, contudo, existem poucos meios para relatar a ocorrência de incêndios. Oficialmente, só existem as torres de vigia florestal com grupos especializados na proteção de florestas. Alguns cidadãos comunicam a deteção de incêndios florestais às autoridades locais servindo

de grande suporte no combate aos incêndios, visto que as torres de vigia, apesar de estarem localizadas de forma estratégica, não têm visibilidade em todo o território português [12].

Não só as áreas científicas referidas e as entidades relacionadas com a gestão e prevenção de incêndios, como a Proteção Cívil Portuguesa (PCP), contribuem e se envolvem no combate aos incêndios florestais. Muita tecnologia está envolvida neste ramo da ciência, nomeadamente as tecnologias Geographic Information System (GIS). Estas tecnologias têm sido muito usadas para estes problemas como meio de suporte e auxílio na tomada de decisões para a gestão de incêndios florestais em qualquer fase de estado: Antes, durante e depois do incêndio [56].

Havendo este tipo de tecnologia, resulta útil o uso da mesma no nosso país e, desta forma, evitar limitar os meios de comunicação para o reporte de incêndios florestais. As tecnologias GIS, atualmente, podem vir a ser muito úteis para detetar este tipo de eventos.

2.2 Sistema FireLoc

A maneira mais evidente de tirar benefício às tecnologias GIS é o desenvolvimento de sistemas de software que permitam facilitar a análise e a representação de eventos que requerem informação geospacial. Atualmente, existem poucas aplicações que usem tecnologias GIS em Portugal que permitam dar testemunho e/ou uma contribuição para a deteção de incêndios. Dito isto, é oportuno destacar o sistema em estudo. Apesar do sistema FireLoc ter um grande avanço de desenvolvimento, ainda se encontra em fases de implementação e testes, pelo que, nesta secção, iremos dar continuidade à descrição do projeto, destacando os elementos aptos para ambiente operacional e elementos ainda em fase de desenvolvimento e refinamento.

Características do Sistema

A inovação do projeto FireLoc permite recolher, processar e validar dados georreferenciados quase em tempo real, o que permite identificar um incêndio totalmente baseado em informações fornecidas pelos cidadãos. Desta forma, podemos destacar os pontos fortes do projeto:

- O sistema permite alertar a ocorrência de um incêndio a partir da informação obtida por *crowdsourcing*. A técnica de *crowdsourcing* é uma técnica inovadora e promissora, uma vez que grande parte da população portuguesa encontra-se nas proximidades do tecido florestal, tornando-os potenciais utilizadores. Qualquer cidadão é um potencial utilizador do sistema o que possibilita gerar um maior resultado a partir do contributo de cada um;
- Torna evidente a localização dos eventos identificados para os utilizadores por meio de visualização desta informação em mapas geográficos;
- O sistema conta com a participação não só do público geral, mas também das autoridades, o que implica que existam vários tipos de utilizador e, conseqüentemente, seja necessário fornecer a cada tipo de utilizador tipos específicos de serviços e informações que vão de encontro com as necessidades de cada um.

O Fireloc deve estar preparado para gerir as limitações típicas em sistemas de georreferenciação, como, por exemplo, a qualidade e fiabilidade dos dados denunciados

pelos utilizadores e os seus dispositivos. Por esse motivo, um dos objetivos é precisamente aplicar módulos que permitam validar os dados para garantir confiabilidade na informação que é apresentada;

- O sistema Fireloc pretende adotar protocolos reconhecidos e recomendados pela Open Geospatial Consortium (OGC). A OGC é uma comunidade mundial comprometida em melhorar o acesso a informações geospaciais ou de localização [15] e tem, como normas, protocolos padronizados que permitem integrar serviços e estruturas de dados utilizados para a gestão de informação georreferenciada [80], de modo a que a aplicação seja interoperável. A adoção destes protocolos fazem com que o sistema seja capaz de se integrar e funcionar com outros sistemas. Contudo, dada a natureza do problema, o sistema em estudo funcionará exclusivamente no território Português, pelo que deverá ser utilizado um Coordinate reference system (CRS) que nos permita usar e transformar dados georreferenciados adequados em Portugal.

Aplicações e/ou Módulos desenvolvidos

Como foi referido no Capítulo 1, o sistema FireLoc tem duas aplicações para uso operacional, uma aplicação móvel e uma aplicação web. A aplicação móvel é destinada para qualquer cidadão português que deseje denunciar ocorrências de incêndios (para facilitar a citação, as ocorrências de incêndios serão designadas como eventos, e os utilizadores que denunciam ocorrências de incêndios serão designados como voluntários). Desta forma, qualquer utilizador poderá contribuir para a deteção de eventos disponibilizando informações úteis para o sistema, como, por exemplo, o local de observação, fotografias do evento, e a orientação em relação ao norte. Esta informação é facilmente recolhida pelos dispositivos móveis através dos recursos nativos dos smartphones. Atualmente, a aplicação móvel consegue recolher dados (contribuições) para que sejam processados pelo sistema. Contudo, esta aplicação já se encontra desenvolvida e, assim sendo, não será alvo de estudo nesta dissertação.

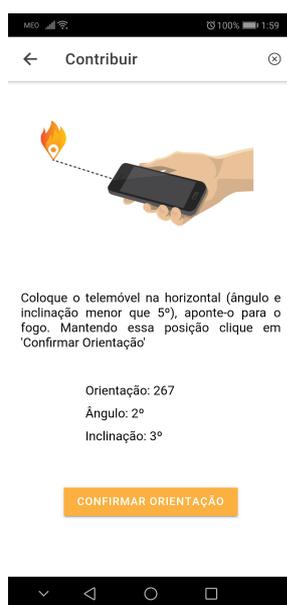


Figura 2.1: Aplicação Móvel - Verificação da Orientação ao Norte

Aplicações e/ou Módulos em desenvolvimento

Apesar de ter referido que a aplicação móvel já se encontra desenvolvida e apta para utilização em situações reais, apresenta uma limitação quanto à sua portabilidade em diferentes Operating System (OS). Atualmente, a aplicação só se encontra disponível para Android e está a ser planeado o desenvolvimento para iOS, de modo que, o sistema se torne mais escalável a nível de negócio e se possa ter mais aderência por parte da comunidade.

Por outro lado, a aplicação web (Portal) é destinada para qualquer utilizador do sistema, nomeadamente, utilizadores comuns, os gestores, que neste caso pode ser a PCP, e os administradores (funcionários especializados do sistema FireLoc). Esta aplicação encontra-se, atualmente, em desenvolvimento, podendo ser dividida em 2 partes, uma destinada para utilizadores comuns onde será feita a representação de eventos, gráficos e outras informações úteis, como, por exemplo, a área do evento em questão, o número de frentes e a sua extensão, a causa de ignição, o estado atual do evento, entre outros. É possível também, aceder às fotografias do evento e filtrar alguns serviços, como, por exemplo, a observação de alguns pontos de interesse (hospitais, escolas ou bombas de gasolina).

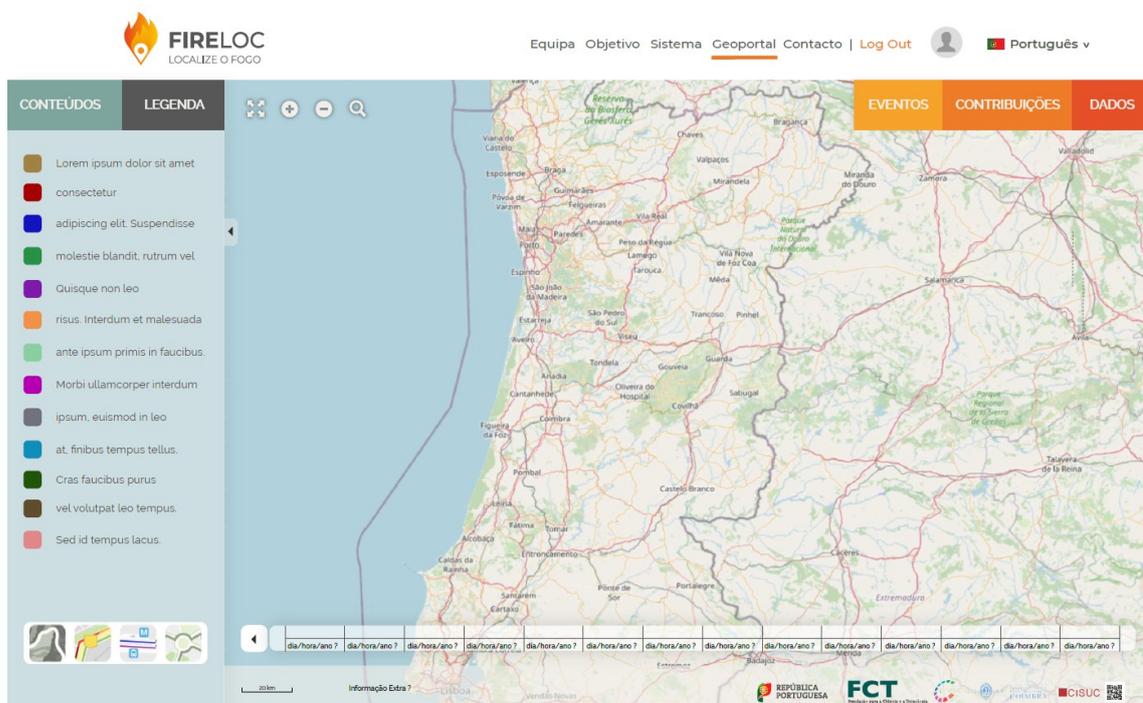


Figura 2.2: Prototipagem da User Interface (UI) Portal - ZIP Design: Creativity Agency

A outra parte é destinada para os gestores e administradores, uma vez que se trata da monitorização de toda a informação do sistema, tanto de análise como de tomada de decisões. Podemos assumir que esta parte da aplicação web é o back-office, o que facilita a agilização de alguns serviços e processos privados para funcionários autorizados.

Quando foram iniciadas as atividades práticas relativamente às funcionalidades, alguns módulos já se encontravam desenvolvidos, nomeadamente, o Sistema de Autenticação, Gestão de Utilizadores, a Gestão de Grupos, a Gestão de Eventos e a Gestão da Grelha de Referência Geográfica. No entanto, a equipa de desenvolvimento testou e analisou possíveis melhorias para otimização das funcionalidades, de modo que se conseguissem detetar falhas na implementação ou nas estruturas dos modelos. Como resultado, detetou-se vários pontos a melhorar em alguns destes módulos, obrigando a fazer uma reestruturação, desenvolver

mais componentes e realizar uma nova fase de testes. Desta forma, irá assumir-se que os módulos mencionados não se encontram desenvolvidos e, por este motivo, serão adicionados aos módulos apresentados e descritos no Capítulo 3, assim como os restantes módulos a desenvolver.

Atualmente, o sistema FireLoc conta, também, com uma equipa que se está a focar especificamente no desenvolvimento de módulos relacionados com a classificação de imagens para que o sistema seja capaz de validar os dados dos utilizadores que contribuem para a identificação e localização de incêndios (voluntários). Ainda, existe uma equipa que está a tratar dos cálculos matemáticos necessários para a identificação de incêndios com alguns dos parâmetros fornecidos pelos voluntários na aplicação móvel. Para que fique mais clara a contribuição no projeto, será apresentada a arquitetura existente do sistema FireLoc e dos principais módulos independentes.

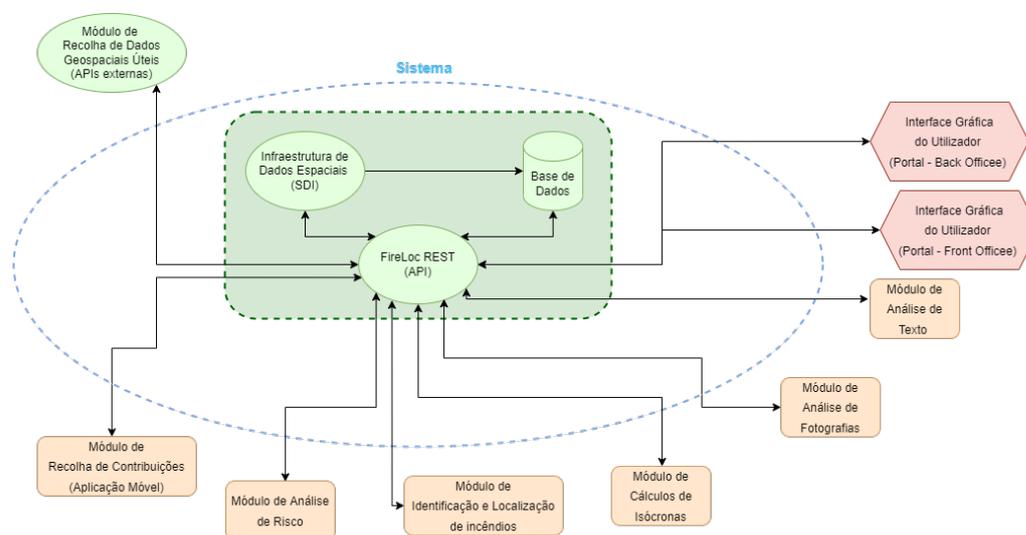


Figura 2.3: Arquitetura do Sistema FireLoc

A figura 2.3 representa uma visão da arquitetura do sistema FireLoc com um alto nível de abstração, na qual são apenas identificados os principais módulos e a Application Programming Interface (API). Ao observarmos a figura, fica clara a importância da API do sistema uma vez que todos os módulos do esquema comunicam diretamente com a mesma.

O foco deste trabalho foram os módulos ovais, nomeadamente, o desenvolvimento de módulos e componentes na API, a interação com a base de dados, atividades com o Spatial Data Infrastructure (SDI) e a extração de dados provenientes de fontes externas. O SDI é um módulo especial e será descrito, neste Capítulo, numa secção mais em frente. Os módulos retangulares são da responsabilidade de outras equipas especializadas do FireLoc, nas quais estão a focar as suas atividades.

Os módulos hexagonais fazem parte das aplicações de cliente do sistema e não serão focos de estudo, assim como os módulos de retangulares.

No Capítulo 4, será feita uma nova representação da arquitetura atual, na qual se irá explicar com mais detalhe os módulos ovais. Nesse capítulo, ficará esclarecida a importância da API e dos módulos que a constituem e que não se encontram visíveis na arquitetura representada na figura 2.3.

2.3 Aplicações existentes

Nesta secção serão identificadas algumas aplicações que integrem, tal como o sistema FireLoc, tecnologias GIS para se poder fazer uma comparação entre sistemas, com o intuito de identificar semelhanças e/ou desigualdades entre elas.

Para que a comparação seja coerente, no contexto em que está inserida, são apenas comparadas aplicações desenvolvidas para uso em território português ou semelhante, pelo que a maioria de aplicações do estrangeiro não serão consideradas.

2.3.1 Aplicação Fogos

Existem aplicações informativas, como Fogos [62], que nos notificam com uma série de informações para localizar incêndios e o estado do mesmo. Esta aplicação conta com o apoio e contribuição de denúncias da Associação de Voluntários Digitais em situações de emergência (VOST) [65]. Contudo, a grande maioria dos dados é fornecido pela PCP.

Atualmente, Fogos conta com 2 aplicações, uma em versão web e outra em versão mobile para facilitar o acesso à informação em qualquer lugar e momento. Estas aplicações são de acesso público sem necessidade de registo. A sua principal feature é a representação de dados geográficos usando um mapa, como se pode observar na figura 2.4.

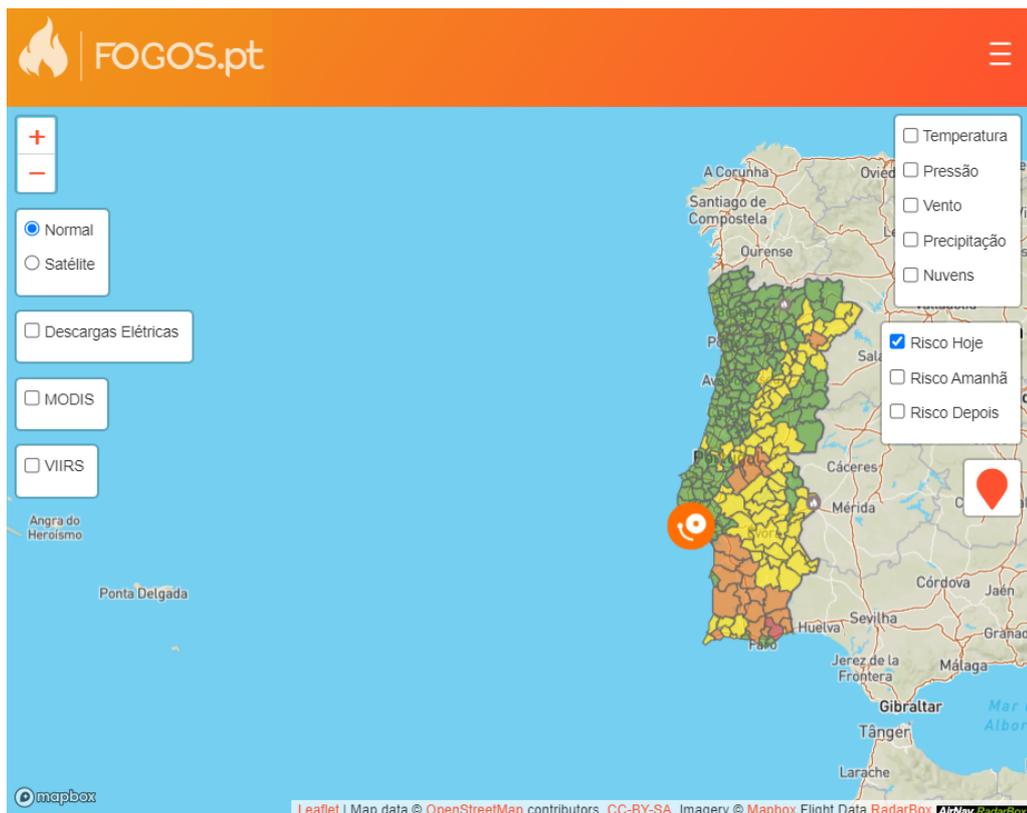


Figura 2.4: O mapa de incêndios ativos no território português é aqui visto em conjunto com a informação da temperatura atmosférica

A informação visível do incêndio no mapa pode ser controlada com os filtros da sidebar direita, assim como algumas informações meteorológicas. A aplicação também possui outros tipos de representação da informação relativa aos incêndios em formato de tabela e

lista que nos informam do estado atual do incêndio, o local, a hora em que se detetou o incêndio e, também, dos meios disponibilizados para o combate do mesmo (meios humanos, terrestres e aéreos). Inclui, também, conteúdo estatístico com diferentes tipos de gráficos e informações, muito úteis para a gestão de incêndios.

A versão mobile é semelhante à versão web, tendo todas as funcionalidades anteriormente referidas, com exceção aos filtros meteorológicos.

2.3.2 Sistema GesFoGO

A Gestão de incêndios Florestais mediante Georreferenciação em Observadores móveis (GesFoGO) é um projeto desenvolvido pela equipa Enterprise Engineering Laboratory (Eelab) em parceria com Universidade de Las Palmas de Gran Canaria (ULPGC) aprovado pelo Programa de Cooperação Territorial INTERREG V A Espanha-Portugal e pelo Interactive Technologies Institute (ITI) [59].

O projeto está focado na prevenção e gestão de incêndios que afetam a maioria das zonas verdes de Canárias, Madeira e Cabo Verde, de forma a melhorar a proteção destas zonas recorrendo a tecnologias observadoras e de custo rentável (drones e sensores implantáveis) envolvendo uma série de entidades como a PCP, Bombeiros de Cabo Verde, o Instituto da Conservação da Natureza e das Florestas (ICNF) e Governos [41].

Este foi um projeto na qual o estagiário participou no passado, tendo estado envolvido numa série de atividades relacionadas com o desenvolvimento de software do sistema proposto, nomeadamente, o desenvolvimento da UI e da API para receção de dados e envio de comandos para os sensores.

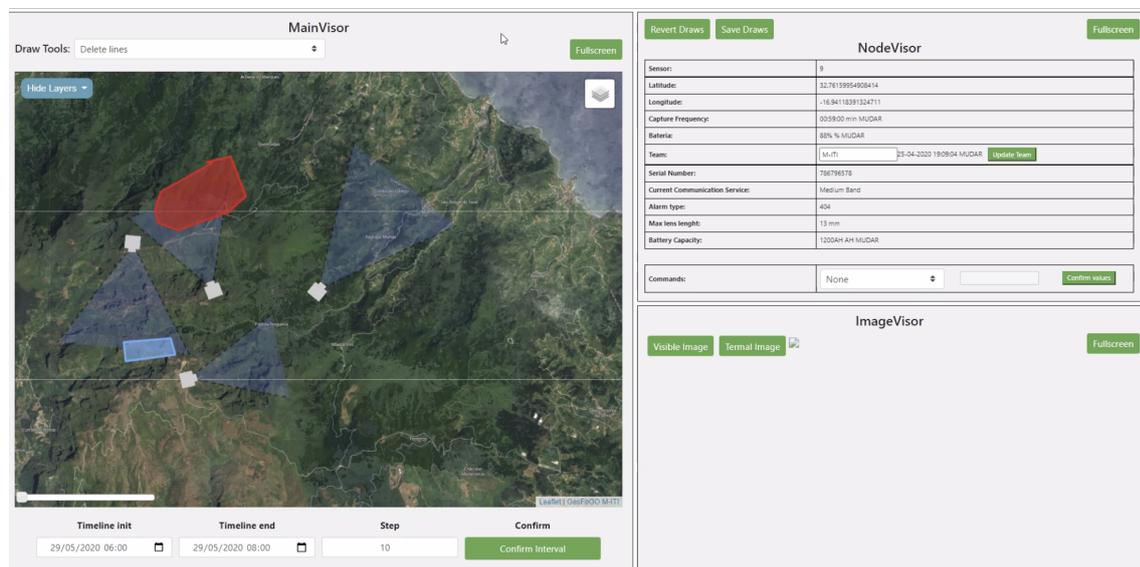


Figura 2.5: Interface Gráfica do Utilizador GesFoGO - Imagem retirada de [75]

Apesar do envolvimento, não é possível partilhar detalhadamente o conteúdo do projeto, pelo que a referência aqui feita é apenas para se exaltar a existência e potencial uso do sistema GesFoGO em Portugal. Alguns pappers foram publicados, abordando o sistema de hardware desenvolvido, visto que a ULPGC é a responsável pelo projeto e pelo desenvolvimento dos componentes físicos que serão usados (drones e sensores implantáveis) [41]. A inovação do GesFoGO é construir um sistema que seja facilmente adaptável para qualquer região com características semelhantes à Macaronésia e poder operar em condições

climatéricas adversas (chuva, vento, entre outros) e nas quais torna-se, muitas vezes, difícil para os meios humanos e/ou terrestres intervir. Isto significa que, por vezes, não se poderá usufruir da operação ativa dos drones pelo que os sensores implantáveis são o principal meio operacional do sistema GesFoGO. Estes sensores estão colocados no perímetro de territórios mais propensos a incêndios e, caso estes se encontrem no meio do fogo, têm sistemas de auto-segurança que lhes permite proteger-se das altas temperaturas sem que a informação seja perdida e/ou o serviço seja interrompido.



Figura 2.6: Processo de deteção GesFoGO - Imagem retirada de [75]

2.3.3 Sistema Bee2Fire

O sistema Bee2Fire foi um projeto desenvolvido em Portugal com o objetivo de detetar incêndios florestais com módulos inteligentes. A aposta desta aplicação é na identificação precoce do incêndio para evitar consequências irreversíveis. Tal como o FireLoc, propõe uma solução única e inovadora [49].

O sistema inclui a instalação de vários sensores no território Português com sensibilidade térmica e com capacidades de espectrométricas para detetar incêndios. Os sensores podem ser instalados em várias condições de terreno, o que facilita a integração deste sistema com outros. Atualmente, é usado em Parques Naturais, Áreas de Mineração e Instalações industriais para vigilância de eventos.

A inclusão de módulos inteligentes permite realizar análises muito úteis para às equipas e entidades competentes como a ICNF e PCP. Entre a informação útil pode-se destacar três categorias: Probabilidade de ocorrência de incêndio, Deteção, e Propagação [49].

- O sistema consegue determinar a probabilidade de ocorrência de um incêndio florestal mediante informação recebida pelos sensores;
- A deteção dos incêndios é feita recorrendo a tecnologias de análise de imagens, como o Visual Recognition IA obtidas dos sensores, e fontes de dados externas;
- Consegue-se estimar a direção de propagação do incêndio, assim como, o seu comportamento ao longo do tempo.

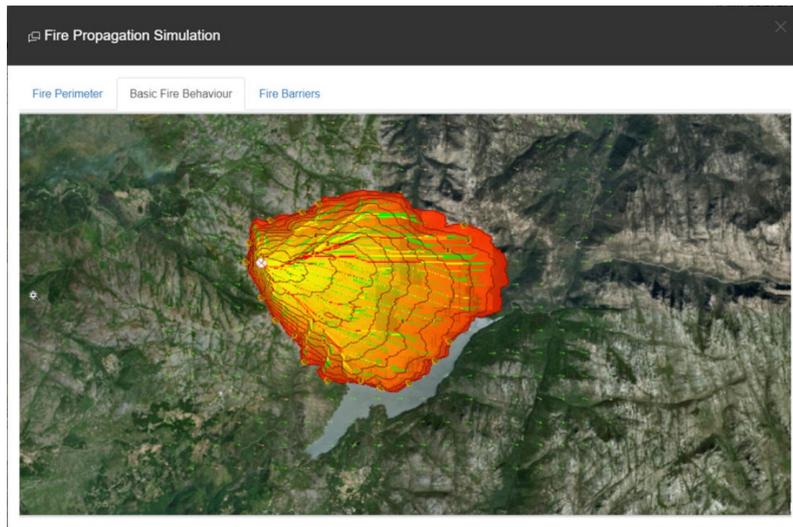


Figura 2.7: Imagem retirada de Bee2Fire -Inteligência artificial [49]

O acesso ao sistema Bee2Fire é feito através de uma plataforma web compatível em diferentes dispositivos, como smartphones. O acesso a determinadas features da aplicação está restringida a tipos de utilizador. Esta solução permite substituir torres de vigia tradicionais assistida por humanos e servir de igual maneira como sistema de apoio à decisão, o que permite uma atuação e gestão rápida dos incêndios florestais.

2.3.4 Aplicação Fire RoS Calculator - Associação para o Desenvolvimento da Aerodinâmica Industrial (ADAI)

É importante destacar, também, o trabalho que é feito em parcerias com a Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC). O Centro de Estudos sobre Incêndios Florestais (CEIF) é uma unidade da ADAI ligada ao Departamento de Engenharia Mecânica da FCTUC que se ocupa na investigação, no campo de incêndios florestais. Estas entidades têm estado envolvidas numa série de estudos e projetos relacionados com esta área da ciência e têm tido bastante reconhecimento, não só a nível nacional, como a nível internacional [60].

Entre as suas atividades de investigação, atualmente, têm cerca de 17 projetos a decorrer no contexto de incêndios florestais e/ou rurais. Estando estes projetos a decorrer, não há muita informação detalhada, que se possa partilhar, sobre estes projetos. No entanto pode-se destacar a aplicação The Fire Rate of Spread Calculator [3] desenvolvida por estas entidades, e que permite simular a taxa de propagação de um incêndio baseada na calibração de imagens captadas por sensores. A calibração dos sensores e das imagens permitem medir distâncias reais e, por sua vez, calcular a taxa de propagação de um incêndio.

Para tal, é disponibilizada uma aplicação desktop com uma UI simples a nível de design mas bastante técnica e complexa no que concerne a sua usabilidade por parte de utilizadores comuns.

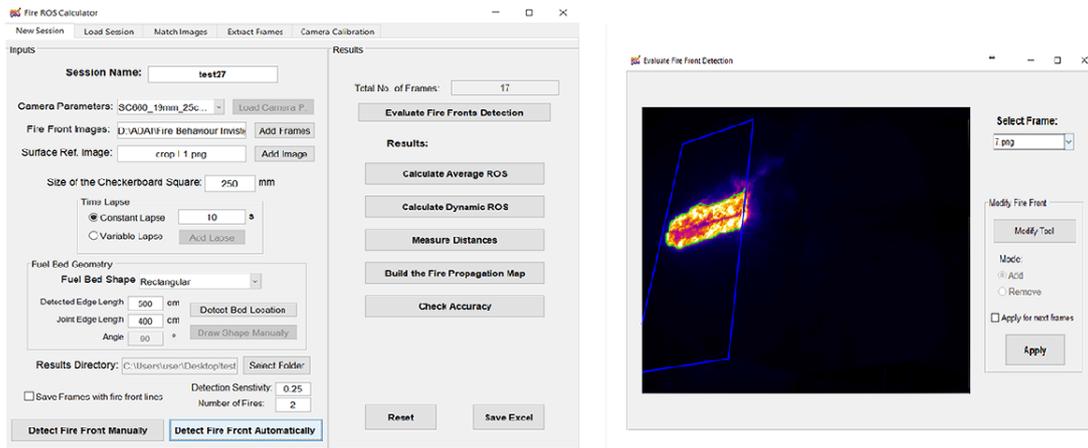


Figura 2.8: Imagem retirada de: A Tool to Measure the Rate of Spread of a Propagating Wildfire in a Laboratory Setting [3]

Como se pode observar na figura 2.8, é possível manipular uma série de parâmetros para devolver resultados variantes, como a taxa de propagação média ou a taxa de propagação dinâmica, o mapa de contorno de propagação do fogo e uma representação 3D da taxa de propagação sobre a superfície do fogo. Esta aplicação parece ter um público alvo muito restrito dada a apresentação dos resultados. Deverá ser destinada para profissionais da área e que visem ter ferramentas para tomadas de decisões. No entanto, esta aplicação não deverá ser utilizada por utilizadores comuns, visto não terem a experiência para poderem tirar o melhor proveito a este tipo de informação.

2.3.5 Sistema IMfire - ADAI

O sistema IMfire é um projeto que está a par em termos de desenvolvimento com o projeto FireLoc e que tem como principal objetivo melhorar os atuais sistemas concebidos para incêndios florestais. O sistema está a ser desenvolvido de forma a que seja altamente interoperável e que seja adaptável a qualquer tipo de região no mundo. O sistema IMfire pretende utilizar diversas fontes de dados para a gestão e prevenção de incêndios florestais, das quais pode-se destacar imagens de satélite, informação recolhida por drones e dados enviados desde o terreno, em conjugação com ferramentas de processamento de módulos inteligentes.

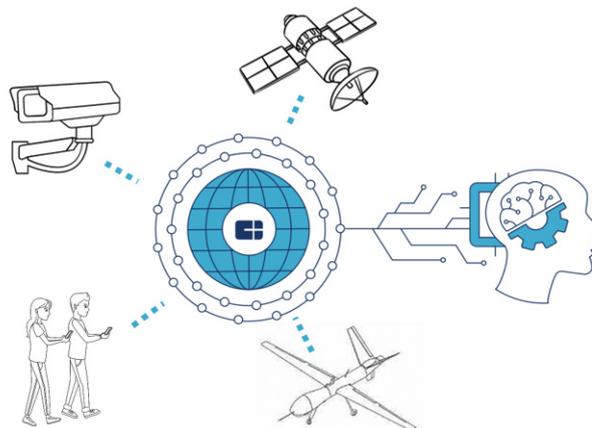


Figura 2.9: Imagem retirada de IMfire - Gestão de incêndios [61]

IMfire disponibiliza os seus serviços através de aplicações web-based e tem, como foi mencionado anteriormente, a intenção de integrar módulos inteligentes para a gestão de incêndios em várias etapas:

- Prevenção - O sistema consegue determinar um conjunto de indicadores de risco de incêndios;
- Planeamento - O planeamento é feito baseado na melhoria contínua das previsões realizadas ao longo do tempo;
- Combate - Análise de resultados para a eficiência das estratégias de combate aos fogos.

Foi realizado, em Maio de 2021, um workshop deste projeto [76], descrevendo o estado atual de desenvolvimento e na qual é mostrado o estado atual da UI, como se pode observar na figura 2.10.

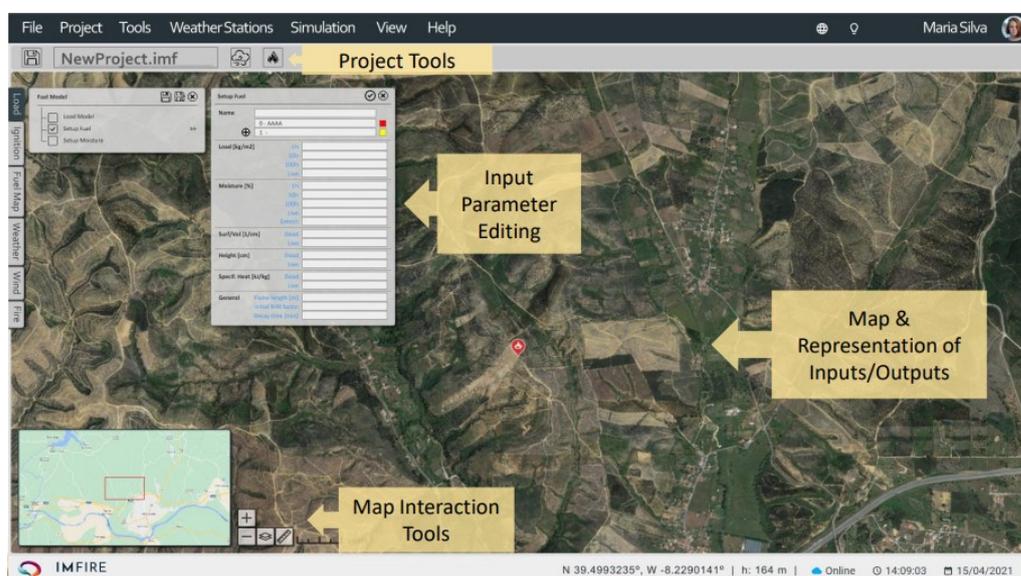


Figura 2.10: Imagem retirada de 2nd Workshop – Scientific Research and Technological Development - Forest Fire Prevention and Fire Fighting [76]

2.4 Comparação entre sistemas

Fazendo uma comparação das aplicações mencionadas com o sistema FireLoc, pode-se identificar alguns pontos que poderiam servir de inspiração ou base de desenvolvimento.

O projeto FireLoc, atualmente, conta com uma série de módulos personalizados e desenvolvidos pela própria equipa, como a gestão de utilizadores, gestão de informação e dados geográficos, avaliação e caracterização da localização de eventos, validação e reconhecimento de imagens carregadas com algoritmos de Machine Learning, processamento e classificação de texto fornecido pelos utilizadores, e um servidor de mapas para permitir que os clientes visualizem informações geográficas.

Contudo, existem variantes destes módulos e que estão presentes nos sistemas apresentados na secção anterior. As funcionalidades mais comuns são a avaliação e caracterização

da localização de eventos, de modo a registrar e denunciar acontecimentos que determinem a estratégia de contenção dos mesmos. Outro módulo muito comum é o servidor de mapas, cada vez mais utilizado em projetos de planejamento e gestão de sistemas com dados geospaciais. Todas as aplicações mencionadas na secção anterior possuem esta funcionalidade. Isto é um indicador forte de que a utilização destas soluções são bastante eficientes e respondem com sucesso aos problemas de gestão e prevenção de incêndios.

A aplicação Fogos possui duas versões de uso, uma em web e outra para dispositivos móveis tal como o FireLoc. Apesar da aplicação Fogos ser apenas informativa, tem algumas semelhanças quanto às funcionalidades do FireLoc.

O FireLoc conta com um módulo específico para a gestão de risco do território nacional enquanto Fogos tem um componente de filtragem que apresenta zonas de risco. O sistema IMfire tem, também, a gestão de risco como uma das suas principais funcionalidades, determinando um conjunto de indicadores de risco de incêndios para a prevenção dos mesmos. Tendo em conta que uma das grandes preocupações e apostas da logística florestal é identificar zonas de risco para a prevenção, faz sentido ter esta funcionalidade/módulo presente em qualquer sistema, algo que já o FireLoc possui e que deve ser realçado.

A aplicação Fogos conta também com um sistema de contribuições, como foi referido anteriormente. Contudo, as contribuições não têm origem pública, mas sim contribuições privadas de entidades selecionadas e reconhecidas para o efeito, como é a VOST. O FireLoc tem um sistema de contribuições público, isto é, qualquer utilizador consegue colaborar na gestão e prevenção de incêndios com a sua contribuição. Estas, por sua vez, são feitas através da aplicação móvel e devem ser processadas para evitar falsos positivos o que torna o sistema de contribuições muito mais complexo, isto significa que, no contexto do projeto, uma contribuição é uma colaboração por parte do utilizador com os seus dados para nos auxiliar na identificação e localização de incêndios florestais. Apesar da complexidade, é muito mais útil em comparação com Fogos pelo facto de se ter uma ampla fonte de dados, os cidadãos. Por conseguinte, ter mais contribuições traduz-se em mais informação e conhecimento e, por sua vez, numa melhor gestão e prevenção. O sistema GesFoGO e Bee2Fire não têm sistemas de contribuições por parte de utilizadores por apenas se basearem em dados sensoriais. Por outro lado, o sistema IMfire da ADAI, apesar de não ter nenhuma referência sobre o uso de contribuições por parte de utilizadores, exploram várias fontes de dados como se pode observar na figura 2.9. O FireLoc, por exemplo, não só se limita a usar apenas contribuições da aplicação móvel, mas também contribuições voluntárias de APIs externas neste contexto, como, por exemplo, o Open Street Map (OSM). O OSM é desenvolvido por uma comunidade voluntária que contribui e mantém atualizados os dados sobre estradas, trilhos, cafés, estações ferroviárias, incluindo ocorrências de incêndios e muito mais categorias, por todo o mundo [78].

Apesar desta grande vantagem, é importante referir que os dados crowdsourced implicam enfrentar alguns problemas de fiabilidade. O FireLoc está preparado para avaliar a fiabilidade de imagens e descrição dos eventos (ocorrência de incêndio), mas, no que concerne a localização dos dispositivos, pode tornar-se um grande desafio. Os dispositivos móveis sofrem grandes desvios no sensor de localização (Global Positioning System (GPS)) por fatores físicos (como, por exemplo, estar próximo de uma linha de água) o que provoca um maior alcance de erro. Outro fator importante é a experiência do voluntário, interesses ou intenções (considerando que, infelizmente, existem indivíduos motivados a prejudicar o bem estar das pessoas e das florestas) [12]. Este ponto de fiabilidade e validação de contribuições por parte dos utilizadores será um tópico abordado no Capítulo 3 com mais detalhe.

Relativamente à interface das aplicações, infelizmente, as UI podem tornar-se muito com-

plexas em termos de usabilidade e comprometer severamente a experiência de utilizador. Na área de informação geográfica existem inúmeros conceitos e abreviaturas que apenas pessoal especializado na área consegue compreender, pelo que é vital que as aplicações apresentem uma UI clara e com boas práticas em termos de usabilidade, evitar conceitos técnicos para utilizadores comuns, entre outros. É evidente que isto dependerá do público alvo que se pretende alcançar. O sistema Fire RoS Calculator da ADAI é um sistema com uma UI muito simples em termos de design, mas em termos de experiência de utilizador apresenta grandes problemas para utilizadores não especializados na área. A UI tem uma grande carga de informação, apresentando uma complexidade elevada de utilização pela quantidade de conceitos e entradas de configuração. Quando se deseja alcançar outros tipos de utilizador, como os cidadãos portugueses, a UI tem de ser desenhada para que este público alvo consiga compreender e utilizar o sistema sem ambiguidades.

Por esse motivo, o sistema FireLoc optou por construir uma UI para a aplicação móvel com guidelines para os utilizadores do sistema e com uma interface menos sobrecarregada de informação. Na verdade, as guidelines têm como objetivo evitar que as contribuições dos utilizadores sejam um problema, minimizando erros que possam ser introduzidos no sistema por parte dos utilizadores e evitar alguns dos desafios de crowdsourcing. Torna-se difícil comparar a UI entre a aplicação móvel Fogos e a aplicação móvel FireLoc por terem propósitos distintos, contudo, é importante realçar que há semelhanças entre ambas.

Por outro lado, as interfaces das aplicações web são mais simples de comparar. A aplicação Fogos, GesFogo, Bee2Fire, IMfire e FireLoc têm mapas em todas as suas aplicações. Todos estes sistemas têm toolbars integradas com os mapas de forma a que as interfaces tenham a área representativa da informação geográfica, mas também das funcionalidades, de forma a que se possa observar em tempo real as mudanças no mapa daquilo que se está a manipular.

Outro ponto de comparação, e que é idêntico nas aplicações mencionadas, é a representação de imagens reais captadas por sensores sejam eles privados, como por exemplo os sensores do GesFoGO e Bee2Fire, ou de acesso público, como, por exemplo, imagens do satélite Sentinel-2 usadas no sistema FireLoc através da API SentinelHub. A exibição destas imagens são uma representação mais realista do foco de observação para que a gestão dos incêndios seja mais apurada.

Comparando as estratégias de identificação e deteção de incêndios, a estratégia que o sistema FireLoc utiliza para a identificação de incêndios é uma estratégia de triangulação entre dois smartphones e o incêndio propriamente dito.



Figura 2.11: Imagem retirada de [24] - Posicionamento

Como já foi referido, os smartphones recolhem a localização aproximada do indivíduo assim como a sua orientação ao norte. Estas informações permitem-nos inferir a localização dos

fogos. À medida que aumentam as contribuições (denúncias pelas pessoas) consegue-se diminuir o erro que está associado às aproximações de localização. A diminuição dos erros está, também, associada a uma comparação entre eventos registados pelo sistema e eventos reais, registados à priori, isto é, registos observados.

A comparação entre eventos do sistema e eventos reais baseiam-se na existência ou não de uma relação entre eles. Um exemplo de que o evento registado pelo sistema FireLoc tem um erro perplexo associado é nos casos em que uma determinada área ardeu no ano anterior e o sistema deteta que no corrente ano voltou a arder. Nestes casos o evento detetado pelo sistema é suspeito, pois é evidente que a área ardida necessita de tempo para se voltar a regenerar.

A comparação com o GesFoGO deverá ser feita apenas em consideração aos problemas que ambos sistemas pretendem resolver e as suas estratégias, uma vez que o GesFoGO não tem disponibilizado nenhuma informação pública do estado atual da aplicação.

O GesFoGO conta com observadores móveis, isto é, sensores colocados em zonas estratégicas no meio das florestas. Estes sensores poderão ser facilmente manipulados pela aplicação do GesFoGO [36]. O desenvolvimento deste sistema apresenta uma estratégia comum de desenvolvimento de longo prazo nas áreas de segurança e proteção ambiental entre as Ilhas Canárias, Madeira e Cabo Verde, e que poderá ser facilmente integrada para qualquer zona florestal que tenha uma orografia semelhante à da Macaronésia, isto é, nuances do relevo da Macaronésia [21]. Este sistema de deteção é, completamente, diferente ao sistema FireLoc. No sistema FireLoc, como já se referiu, não é utilizada qualquer tecnologia móvel georreferenciada como drones a operar à distância no terreno, mas é utilizada a tecnologia móvel do próprio utilizador, os smartphones. Na prática, os dados são fornecidos pelos utilizadores e pelo próprio dispositivo móvel (localização do dispositivo) e que, por sua vez, são processados pelas API do FireLoc para deteção e localização dos incêndios.

Se compararmos não só o GesFoGO com o FireLoc mas também com o Bee2Fire, o GesFoGO e o Bee2Fire são bastante semelhantes. Ambos utilizam sensores colocados de forma estratégica no terreno e a partir deles recolhem informação. As duas aplicações têm módulos inteligentes que permitem analisar a propagação de um incêndio ao longo do tempo. Estas features são, como já foi referido, muito úteis para as entidades competentes. A comparação entre o Bee2Fire e o FireLoc acaba por ser bastante semelhante à do GesFoGO, tendo sido discutido grande parte dos pontos comparativos. Infelizmente, não será feita uma comparação mais profunda em termos tecnológicos e da aplicação entre o GesFoGO e o Bee2Fire dada a situação de sigilo que o GesFoGO conserva até ao momento.

2.4.1 Sumário

Feita a análise e respetiva comparação das aplicações, depreende-se que são diversas as tecnologias e estratégias que se podem seguir para regular e, simultaneamente, antecipar medidas de contenção para incêndios florestais. Com este estágio, é expectável que sejam melhorados alguns aspetos da aplicação FireLoc na vertente de back-end tendo em conta o que foi discutido.

Visto que o FireLoc já possui uma quantidade considerável de módulos, é de ressaltar a importância da forte integração que o sistema deve ter para que funcionem em conjunto num só sistema.

Deve-se ter em conta que terá utilizadores variados e de qualquer tipo, como cidadãos portugueses e/ou entidades competentes e profissionais. Por esse motivo, deve possuir

uma interface que permita aos utilizadores realizar as suas atividades com clareza, isto é, os conceitos e informações que são representados no FireLoc devem estar bem descritos, assim como, cada ação do utilizador, para evitar ações erróneas.

Algo que se notou facilmente em algumas aplicações é a quantidade de informação apresentada. Alguns conceitos para a maioria dos utilizadores do FireLoc seriam abstratos. Desta forma, deve-se realizar uma análise cuidada e minuciosa da informação que se pretende representar, assim como ter em consideração a forma como esta será interpretada pelos mesmos.

2.5 Tecnologias

Como já foi referido previamente, o FireLoc já se encontra num estado avançado de desenvolvimento no qual o planeamento do mesmo se encontra concluído. Contudo, vale a pena fazer uma revisão das tecnologias usadas no projeto, assim como algumas ferramentas que poderão ser integradas na fração de desenvolvimento que o estagiário estará responsabilizado. Desta forma, podemos dividir as tecnologias em cinco vertentes, tecnologias de web-mapping, tecnologias de front-end, tecnologias de back-end, base de dados e virtualização.

Pretende-se, nesta secção, reforçar e justificar a utilização das tecnologias pre-selecionadas neste projeto e analisar a viabilidade de outras ferramentas úteis para uma possível integração à lista de tecnologias em uso, caso se apresentem necessidades de utilidade.

2.5.1 Tecnologias em Web Mapping

As tecnologias de web mapping são serviços e/ou processos que conseguem partilhar, estruturar e representar objetos para análise de dados georreferenciados. A principal tecnologia envolvida no projeto FireLoc são as de web mapping e podem dividir-se em diferentes categorias uma vez que a área não se limita a conceituação exata de mapas, mas sim de diversas vertentes como tipos de dados, protocolos, geradores de serviços, entre outros.

Para condensar o conceito de web mapping podemos introduzir um novo conceito, SDI. O SDI representa toda a infraestrutura de dados geoespaciais e pode ser definida como um conjunto de ferramentas, políticas e padrões. Define como adquirimos, processamos, distribuimos, usamos, mantemos e preservamos os dados geoespaciais [82].

O SDI do FireLoc desvia-se do conceito original e inclui, atualmente, tecnologias para gerar serviços de mapas por meio de um servidor (GeoServer), que iremos abordar mais adiante, assim como as interfaces de catálogo. Desta forma, o SDI não inclui a parte de visualização de dados geoespaciais – renderização de mapas. As interfaces de catálogo não são uma prioridade atual do sistema FireLoc, devido a estarem diretamente relacionadas com a parte mais administrativa do sistema como irá ser discutido, posteriormente.. Nesta secção, serão abordadas as principais classes de web mapping, para que se consiga ter uma visão clara do uso das tecnologias do SDI e tecnologias de renderização de mapas [85].

Protocolos OGC

Uma das categorias mais importantes de uma SDI tradicional é a estrutura utilizada e os tipos de dados associados. Esta vertente está diretamente relacionada com a área geomá-

tica. A geomática é um ramo da informática que procura integrar tecnologias e protocolos para aceder e gerir dados geoespaciais, através do uso de serviços de tecnologia e informação georreferenciada [80]. Dito isto, explicar-se-á, nesta secção, o que são os protocolos da OGC, para que são usados e porquê são importantes. Irá explicar-se, também, que tipos de dados estão associados a cada protocolo e quais deles estão a ser utilizados no sistema FireLoc.

As tecnologias GIS têm uma estrutura e tipos de dados associados. Estas estruturas podem ser vistas como serviços, podendo-se destacar as mais comuns: Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), entre muitos outros. Estes três protocolos são adotados pelo OGC e têm diferentes tipos de utilidade, dependendo das necessidades específicas de localização e tecnologia geoespacial interoperável [16]. Muitos destes protocolos são interpretados e suportados por tecnologias de web mapping, sendo um dos critérios de escolha para a grande maioria das tecnologias deste capítulo, ou seja, tecnologias que conseguissem integrar estes serviços. Têm uma importância bastante considerável uma vez que são eles que conseguem tornar o sistema interoperável.

Protocolo WMS

O protocolo WMS é um serviço que permite, através de uma interface Hypertext Transfer Protocol (HTTP), realizar pedidos a um gerador de serviços (SDI do FireLoc) e obter os dados em formato de imagens (imagens georreferenciadas) [14]. Estas imagens podem ter formatos como PNG, JPEG ou TIFF.

Protocolo WFS

Por outro lado, o protocolo WFS permite realizar operações “Create, Read, Update and Delete (CRUD)” sob os dados do servidor através de uma interface HTTP. Estas operações permitem controlar a fonte dos dados georreferenciados assim como características e atributos associados aos mesmos através de queries parametrizadas [13]. Os dados devolvidos podem ter diferentes formatos uma vez que o controlo é sob dados vetoriais. Desta forma, os dados podem ter formatos como GeoJSON ou ShapeFiles. O conceito de imagens geoespaciais e de dados vetoriais é vago, pelo que a melhor maneira de explicar o que estes serviços oferecem é através da figura seguinte.

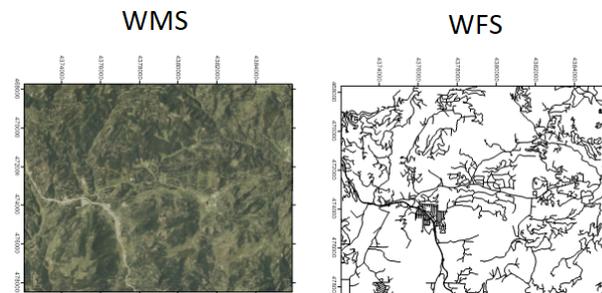


Figura 2.12: Comparação entre WMS e WFS [Adaptado de [69]]

Podemos notar que os serviços são completamente distintos, mas não fica claro o que são imagens geoespaciais em WMS e dados vetoriais em WFS. As imagens geoespaciais são, na verdade, o mapa propriamente dito. Quando estamos a navegar num mapa estamos na verdade a observar um conjunto de imagens geoespaciais. Por outro lado os dados vetoriais podem ser de vários tipos, podendo-se destacar os seguintes:

- Point - Os pontos são uma noção primitiva que modela uma localização exata num dado espaço;
- Line - As linhas são um conjunto de pontos conectados por segmentos de linha;
- Polygon - Os polígonos são um conjunto de linhas conectados para formar uma cadeia poligonal fechada.

Estes são apenas os tipos mais primitivos e comuns. Neste sentido, o protocolo WFS devolve este tipo de objetos que podem ter significados num determinado contexto. Caso se opte por não usar o protocolo WMS, fica a ideia de que é um mapa vazio como se pode observar à direita, na figura 2.13. Por esse motivo, é pouco comum não se utilizar ambos os serviços conjuntamente.

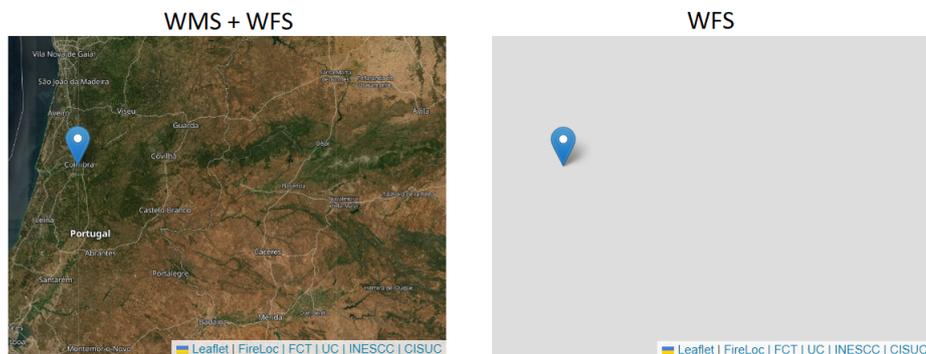


Figura 2.13: Resultados dos serviços WMS e WFS

Protocolo WCS

Pode fazer sentido em determinados projetos GIS ter acesso a informações mais expressivas para fazer uma análise e gestão mais apurada. O padrão WCS pode ser útil nestes casos, uma vez que permite ter acesso à cobertura das imagens. A cobertura pode ser variada dependendo do que se pretende observar. A variação de temperatura ou altitude são alguns exemplos de cobertura que podem ser interpretados em imagens, mas o que realmente interessa são os valores que contém.

Enquanto que no WMS devolvemos imagens para exposição sem valores associados, no WCS devolvemos imagens com coberturas digitais e os seus valores, isto é, imagens que podem ser interpretadas através dos seus valores e não apenas retratadas [84]. Uma outra grande diferença entre os serviços é que o WCS particiona a imagem, isto significa que não devolve uma única imagem, mas sim um dataset.

Se compararmos o WCS com o WFS, em termos da manipulação e interpretação dos dados são também semelhantes. Isto faz-nos refletir e inferir que WCS está perto de ser uma mistura de ambos WMS e WFS num único serviço.

Tipos de dados

Como já foi referido, cada protocolo tem tipos de dados associados. Alguns deles já foram mencionados, contudo, com nomenclaturas incorretas no contexto em estão a ser tratadas. Até agora, o conceito de imagem tem sido usado para facilitar a compreensão dos conceitos

GIS, contudo, o nome adequado na geomática é raster. Por esse motivo, a partir de agora, as imagens devolvidas pelos serviços serão referidas como raster.

Raster

Os dados raster são muito importantes para o sistema FireLoc, principalmente para a identificação dos incêndios florestais. Infelizmente não existe nenhuma API pública que nos permita aceder a uma série de informação útil do país para o sistema realizar a identificação dos incêndios de forma mais apurada. Entre esta informação útil, podemos destacar a Cartografia de Uso e Ocupação do Solo (COS) disponibilizada pela Direção Geral do Território (DGT), na qual podemos identificar em cada lugar do território se se trata de uma zona urbana, se é uma floresta, entre outros. A partir desta informação, é possível, também, determinar outro tipo de indicações como distâncias, elevação do terreno, existência de linhas de água, entre outros. Estes dados existem, mas não podemos acede-los numa API pública, pelo que foi necessário desenvolver um módulo específico para a gestão de raster datasets, como iremos introduzir no Capítulo 3 e descrever no Capítulo 5.

Estes tipos de dados têm propriedades muito específicas, tal como qualquer dado geoespacial. Os rasters consistem numa matriz de células (pixeis), em que cada uma delas, tem informação associada. Esta informação dependerá, como já vimos, do tipo de serviço associado, WMS ou WCS. Usando rasters como mapas base (WMS) poderemos retirar informação relevante como o tamanho da célula. A partir dessa informação, é possível calcular as distâncias entre os seus vértices, correspondentes às distâncias reais e que, por sua vez, permite recolher informação útil para a identificação dos incêndios.

Por outro lado, podemos utilizar rasters como mapas de superfície, ou mais conhecidos como um Modelo Digital de Elevação. Este modelo não é, na verdade, um raster mas sim uma representação dos valores da matriz desse raster, como se pode observar na figura 2.14. Estes valores representam, por exemplo, a temperatura, concentração de densidade populacional, entre outros.

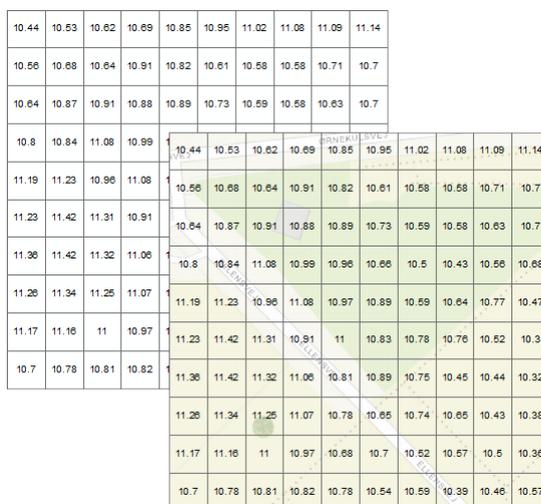


Figura 2.14: Raster em Modelo Digital de Elevação [6]

Vetorial

Os dados vetoriais são um tipo de dado geospacial muito comum e talvez o mais usado para a representação de objetos geospaciais. Alguns deles já foram referidos como, Pontos, Linhas e Polígonos. Estes dados são especialmente úteis para o sistema FireLoc uma vez que nos permitem representar pontos de interesse, focos e contribuições (Pontos), percursos e trilhas (Linhas), e frentes de incêndio, áreas ardidadas (Polígonos), entre outros. Estas geometrias devem ser manipuladas e processadas antes de poderem ser usadas.



Figura 2.15: Dados vetoriais - Features

Como já foi visto, os dados vetoriais são, na sua versão mais primitiva, um conjunto de pontos, pontos esses traduzidos para coordenadas geospaciais (latitude e longitude). As coordenadas dependem do CRS, uma estrutura usada para medir, com precisão, locais na superfície da Terra [83]. Cada país tem o seu próprio CRS baseado num código European Petroleum Survey Group (EPSG). Estes códigos são projeções para o CRS adequado, servindo de referência para um dado sistema de coordenadas, sendo, no caso de Portugal, o EPSG de 3763.

Desafios associados aos dados geospaciais

Em resumo, a informação geospacial pode ser definida como qualquer tipo de informação que tem algum tipo de vínculo geográfico e que faz referência a uma localização na Terra. Pode, como vimos, tomar vários tipos: Raster/Imagens (Mapas), Pontos, Territórios (Polígonos), entre outros.

Trabalhar com dados geospaciais não é comum na área de Engenharia de Software e tem sido um dos grandes desafios desta dissertação, principalmente por toda a manipulação que tem de ser realizada, como o processamento para que, efetivamente, possam ser armazenados estes dados no sistema e, posteriormente, utilizados em ambiente operacional. Desta forma, podemos identificar alguns desafios comuns ao trabalhar com dados geospaciais.

No que toca aos dados raster, o tamanho das células determinam a resolução na qual os dados são exibidos e isto provoca que a representação adequada de algumas características do raster seja difícil de aplicar, dependendo da resolução da célula. A quantidade de células, contudo, pode variar dependendo do nível de zoom que se está a definir no mapa, o que implica que sejam aplicadas estratégias de renderização dos mapas para que a informação seja comprimida para a área em foco e evitar que os dados sejam carregados na totalidade.

Para além disso, existem, também, desafios no processamento de dados vetoriais. Estes dados, como já foi mencionado, têm uma geometria associada, acompanhada de uma projeção EPSG. Deve-se projetar os dados para um EPSG adequado, uma vez que, se não for aplicado corretamente, a representação dos dados será incorreta. Por vezes, as geometrias podem ser Multi Polígonos, que são como o próprio nome indica, um conjunto de polígonos. Nestes casos, é necessário processar cada um dos polígonos que constituem o objeto, o que por vezes pode tornar-se um processo bastante demorado e complexo.

O sistema FireLoc utiliza várias APIs externas para obtenção de dados geospaciais que irão ser abordadas mais adiante. Utilizar várias APIs diferentes torna o desenvolvimento de scripts de automatização do processamento e extração dos dados de interesse bastante complexo, pois todos têm estruturas muito diferentes e específicas. Tem sido um grande desafio adaptar e/ou criar novos scripts para cada uma das APIs.

Para superar os desafios e facilitar o processamento dos dados geospaciais, são utilizadas algumas bibliotecas específicas para os problemas que foram descritos nesta secção. Entre as bibliotecas destacamos o Geospatial Data Abstraction Library (GDAL), considerada uma biblioteca imprescindível para gestão de dados geospaciais, uma vez que fornece ferramentas muito importantes para leitura de dados raster e vetoriais, transformações e projeções, entre outras funcionalidades. A biblioteca Geographic Resources Analysis Support System (GRASS) também é utilizado para a gestão de dados geoespaciais e análise, processamento de imagens, gráficos e mapas de produção, modelagem espacial e visualização [79]. Ambas as bibliotecas são complementares uma com a outra.

Renderização de Mapas

A renderização de mapas propriamente dita é feita através da incorporação de aplicações do lado do cliente para representação de objetos georreferenciados. Os protocolos da OGC descritos na secção anterior interpretam dados geoespaciais e constroem serviços para que as tecnologias de renderização de mapas traduzam o seu conteúdo na representação visual de objetos geospaciais em mapas [77]. Isto implica que a tecnologia escolhida para representar a informação suporte os protocolos da OGC.

Atualmente, a maioria das pessoas utiliza o Google Maps como serviço do lado do cliente para a renderização de mapas. A API do Google Maps oferece grande performance, mas não é flexível, ou seja, só permite usar e aceder aos serviços do Google Maps.

Apesar da sua popularidade de utilização, o acesso à API do Google Maps para desenvolvimento não é gratuito e apresenta limitações na integração de API externas [38] [63]. A necessidade de esquivar estas limitações, levou à identificação de outros serviços de renderização como o OpenLayers [54], Leaflet[4] e MapBox [50].

OpenLayers

OpenLayers é uma das bibliotecas mais utilizadas para projetos que incluem GIS, possuindo uma comunidade vasta e inúmeras features na sua documentação. É caracterizada pelo alto desempenho que apresenta e pela sua facilidade de integração com outras tecnologias de web mapping que usem protocolos da OGC [54]. Tem módulos de integração para aplicações móveis com acesso aos sensores dos dispositivos e tem uma quantidade de ferramentas muito extensa, o que permite, para sistemas muito complexos ou específicos, conseguir cobrir os requisitos.

Leaflet

Leaflet é, também, uma biblioteca muito usada em GIS com destaque no suporte para aplicações móveis e com a grande quantidade de plugins extensíveis que podem ser integrados. O Leaflet tem uma vasta e ativa comunidade sendo, por isso, mais popular e escolhida pela maioria dos programadores e experientes em tecnologias GIS. Apesar de suportar protocolos da OGC como WMS de forma nativa, e, também, por meio de plugins, não

suporta WFS. Contudo, existem plugins externos que permitem a integração deste protocolo, tornando a library completamente interoperável. Uma das suas principais vantagens em relação às outras bibliotecas é ser *lightweight* [4], isto é, na sua versão mais simples, possui as ferramentas mais importantes para representação de informação geospacial e, como resultado, ocupa menos espaço que o OpenLayers, por exemplo. Dependendo do tipo de projeto e dos seus requisitos, esta biblioteca poderá tornar-se mais completa e, conseqüentemente, mais complexa, com a inclusão de plugins. A flexibilidade de integração de plugins permite a equipa de desenvolvimento gerir as ferramentas que necessita e evitar usar tecnologias complexas que contêm ferramentas desnecessárias.

MapBox

Por último, o MapBox é uma biblioteca sem dúvida muito promissora, que oferece uma quantidade de features personalizáveis, com destaque para manipulação de objetos vetoriais. Esta biblioteca tem capacidade para a renderização de mapas em 3D (3 dimensões) e mecanismos de renderização de alta performance para grandes volumes de dados [71]. Permite, também, a integração dos seus mapas em aplicações móveis de uma forma muito simples, tendo features específicas para a construção de aplicações nativas para diferentes OS como Android e iOS [50].

Para facilitar a comparação entre as bibliotecas, selecionaram-se alguns critérios que devem ser satisfeitos para a escolha do melhor serviço de renderização de mapas para o projeto FireLoc. A comparação tem em conta a cobertura dos requisitos do projeto e algumas métricas quantificáveis, assim como, a adoção de práticas e de protocolos interoperáveis.

	OpenLayers	MapBox	Leaflet	Google Maps
Open Source	✓	✓	✓	×
Suporte dos padrões da OGC	✓	✓	✓	✓
Compatibilidade em diferentes OS	✓	✓	✓	✓
Ampla Comunidade ¹	8.9K	8.5K	33.6K	2.5K
Familiaridade	×	×	✓	×

Tabela 2.1: Comparação entre os serviços de renderização de mapas

Feita uma descrição e comparação das tecnologias de renderização de mapas, acabou por escolher-se o Leaflet. Apesar de ser muito semelhante ao OpenLayers e ao MapBox, a familiaridade e a propriedade de *lightweight* do Leaflet foram fatores decisivos. Como podemos observar na tabela 2.1, entre as quatro bibliotecas, o Leaflet destaca-se por ter mais reconhecimento na comunidade, apresentando cerca de 33.6 mil estrelas no GitHub [73], sendo uma métrica bastante útil para compreendermos se um determinado projeto tem popularidade e/ou usabilidade na comunidade. O tamanho das bibliotecas não foi um critério de comparação por ser, de certa forma, uma métrica discutível, visto que podemos inferir que, quanto maior seja o tamanho da biblioteca, maior será o número de funcionalidades que nos outorgará. No entanto, para o sistema FireLoc, é mais viável uma biblioteca *lightweight* pois, com esta característica, nos proporciona todas as ferramentas básicas necessárias para cobrir os requisitos. Caso venha a ser necessário, poderá adicionar-se mais funcionalidades com os plugins que a biblioteca permite integrar. Em relação à familiaridade, com o Leaflet evita introduzir tecnologias novas ou desconhecidas e despender esforço desnecessário na aprendizagem de ferramentas.

¹GitHub Stars - Valores retirados dos repositórios do Github em [57], [51], [1], [37], respetivamente, no dia 14/04/22.

Gerador de serviços web-mapping

Como foi referido anteriormente, o FireLoc optou por seguir uma SDI singular, constituída, principalmente, por tecnologias que permitam gerar serviços web mapping. Foi também referido que uma das vertentes mais importantes de uma SDI tradicional é a estrutura utilizada e os tipos de dados associados e, por essa razão, serão fatores muito importantes a ter em conta nesta secção.

Para compreender-se o que é, na realidade, um gerador de serviços, é importante fazer referência a alguns conceitos já descritos, como, por exemplo, os protocolos da OGC (WMS, WFS, entre outros) e os tipos de dados (Raster e Vetorial).

Um gerador de serviços web mapping é, na prática, um servidor que permite processar, visualizar e editar dados geoespaciais, integrando diversos repositórios de dados geográficos [66]. Desta forma, pode-se gerar um serviço capaz de representar um mapa geoespacial a partir de um repositório de dados georreferenciados, sendo estes repositórios fontes de dados Raster e/ou Vetoriais.

Serviços gerados

Na figura 2.13 pode-se observar como é possível representar a informação através de alguns protocolos da OGC. No entanto, a utilização destes protocolos implica a realização de pedidos HTTP a um servidor especificando o protocolo. Ao realizarmos o pedido ao servidor, este interpreta o protocolo da OGC e, juntamente com os dados geoespaciais, produzem uma referência a um mapa geoespacial. Esta referência ao mapa geoespacial irá ser, por sua vez, utilizado pelo Leaflet para o representar (renderizar a informação geoespacial) na web. Na figura 2.16, podemos observar a interação entre o Leaflet e um gerador de serviços web mapping.



Figura 2.16: Interação entre o Gerador de Serviços e Leaflet

Foi selecionado o GeoServer e o MapServer como principais candidatas a integrarem o SDI do FireLoc. Tal como na escolha das tecnologias de renderização de mapas, também se estabeleceram alguns critérios de seleção de modo a garantir os requisitos e a qualidade pretendida do projeto, nomeadamente, o suporte aos padrões da OGC, a interpretação de diferentes tipos de dados, e as funcionalidades que outorgam.

O GeoServer e o MapServer são ambos servidores web-based para desenvolvimento de aplicações e projetos GIS com o intuito de permitirem a partilha de dados geoespaciais. São ambos utilizados para o desenvolvimento de mapas dinâmicos e personalizáveis, ou

seja que permitem a manipulação de informação geospacial em termos de acesso e como é que essa informação é mostrada através de pedidos HTTP nas suas APIs. Apesar das suas semelhanças, distinguem-se pelos protocolos e os tipos de dados que cada um suporta. Estes servidores são extremamente úteis para sistemas que envolvam informação geospacial por reduzir a complexidade e esforço despendido no desenvolvimento de soluções.

MapServer

O MapServer utiliza MapFiles – «The Mapfile is the heart of MapServer» [52] – como o principal ficheiro de configuração do servidor. No MapFile é especificado como os dados serão lidos e interpretados definindo, desta forma, o tipo de dado que irá ser tratado (Raster, por exemplo), do tipo de serviço em questão (WMS, por exemplo), fontes de dados, entre outros. A dependência do MapFile torna o processo de desenvolvido mais complexo.

GeoServer

Por outro lado, o GeoServer destaca-se por separar o ambiente operacional, baseando-se em três categorias para a gestão e configuração do servidor, nomeadamente, os Workspaces que permitem controlar qual e que tipo de informação está a ser tratada. Na associação desta informação surge outra categoria, as Layers. As layers são a informação vetorial ou raster propriamente dita e que por sua vez estão associadas a uma fonte de dados geospacial. Esta fonte de dados é chamada Store e é, na prática, uma referência da localização do armazenamento dos dados que o GeoServer pode aceder. As Stores podem conter dados raster ou vetoriais com origem numa base de dados ou de ficheiros. É importante referir que o GeoServer e o MapServer não armazenam dados, apenas os acedem e os manipulam.

Em comparação com o MapServer, o GeoServer suporta uma quantidade, relativamente, superior de tipos de dados e oferece mais funcionalidades. Por este motivo, a adesão do MapServer pela comunidade não é tão ampla como a do GeoServer [70]. Contudo, o MapServer tem mais performance que o GeoServer devido às linguagens de programação às quais foram escritas, C e Java, respetivamente.

	MapServer	GeoServer
Open-source	✓	✓
Suporte dos padrões <i>OGC</i>	✓	✓
Compatibilidade em diferentes OS	✓	✓
Ampla Comunidade ²	781	2.7k
Performance ³	0.4s - 0.6s	0.6s - 1.3s
Familiaridade	×	✓

Tabela 2.2: Comparação entre os servidores de que geram os serviços de dados geospaciais

Feita uma descrição e comparação das tecnologias que geram os serviços de dados geospaciais, acabou por escolher-se o GeoServer. Na tabela 2.2 é possível observar que o MapServer apresenta maior performance de renderização comparativamente o GeoServer. Contudo, em termos de reconhecimento, o GeoServer destaca-se pela integração de uma

²GitHub Stars - Valores retirados dos repositórios do Github em [53], [35], respetivamente, no dia 15/04/22.

³Os valores foram retirados de [70]

maior quantidade de tipos de dados e pela sua organização em termos de acesso e gestão de dados geospaciais, o que se traduz numa maior adesão por parte da comunidade.

Aplicações de catálogo

Relativamente aos catálogos do SDI do sistema FireLoc, estes ainda não se encontram integrados, mas poderão ter relevância no futuro e algo a ter em conta no projeto. As aplicações de catálogo são interfaces pré-desenhadas para consultar dados e metadados de informação geoespacial. Estas interfaces podem ser facilmente integradas com os servidores para gestão de informação geoespacial sendo, no caso do FireLoc, o GeoServer. Apesar do FireLoc querer, e ser conveniente, ter o próprio portal de back-office, estes catálogos podem ser muito úteis para controlar e fazer uma melhor gestão dos dados do GeoServer pelos administradores e gestores da aplicação FireLoc.

Os catálogos são, também, muito mais simples do ponto de vista de administração e gestão dos dados em termos de usabilidade e reduzem o custo de desenvolvimento de uma aplicação que faça este tipo de gestão [68].

Deste modo, selecionaram-se aplicações que pudessem oferecer serviços de catálogo no contexto de administração de dados geospaciais. Os candidatos para a integração das aplicações de catálogos são o GeoNode e o GeoNetwork. A aplicação que mais se adequa ao sistema FireLoc, estará sujeita a experiências para uma possível integração. Todas as experiências feitas serão descritas no Capítulo 5.

O GeoNode e o GeoNetwork são Content management system (CMS) geoespaciais que permitem, assim, catalogar informação geoespacial e criar um portal para visualização e gestão da informação seguindo os padrões OGC.

GeoNetwork

O GeoNetwork, em particular, tem uma grande flexibilidade para a configuração do catálogo em aspetos de segurança e visualização dos dados. Este último ponto é relevante devido à grande barreira que existe entre o servidor e a aplicação de catálogo graças a sua estrutura interna. A comunidade que integra o GeoNetwork muitas vezes aproveita esta barreira para re-desenhar o front-end do catálogo e personalizar o seu próprio portal [25].

Assumindo a utilização da interface padrão do GeoNetwork, esta permite visualizar e gerir os meta-dados georreferenciados em tabelas, de forma individual ou aglomerada [43].

Possui funcionalidades de filtragem avançadas sob os dados e também ferramentas para visualização de recursos se estiverem associados a padrões OGC, isto significa que, se os dados do servidor respeitarem estes padrões, podem ser carregados num mapa para visualização dos dados [43].

O grande destaque está na integração de funcionalidades que permitam interagir com o lado do servidor de uma aplicação (back-end). Isto significa que é facilmente integrável em qualquer sistema que utilize Representational State Transfer (REST) APIs. Estas funcionalidades são muito úteis para separar o back-office para múltiplas organizações [43].

O GeoNetwork possui, também, funções para importar e exportar informação de forma manual, assim como, gerar relatórios personalizados para reportar mapas e/ou informação sobre os dados [43].

GeoNode

Relativamente ao GeoNode, este CMS possui muitas funcionalidades do GeoNetwork. Por essa razão serão apenas abordadas as funcionalidades e características exclusivas do GeoNode.

O GeoNode divide a sua aplicação em três partes, cada uma destinada para utilizadores comuns, gestores do sistema e administradores, respetivamente. Desta forma, consegue ter permissões de visualização para diferentes tipos de utilizadores.

As versões atuais do GeoNode adotaram seguir algumas tecnologias open-source como Python (Django), GeoServer, PostgreSQL (PostGIS), entre outros. Apesar de não ser tão flexível como o GeoNetwork na integração da aplicação, garante segurança e a auto-sustentabilidade da aplicação a longo prazo. O FireLoc utiliza as tecnologias mencionadas no projeto, não descartando, assim, a utilização deste CMS [27].

A interface que o GeoNode dispõe para gestão e administração depende da operação que se pretende realizar. Para fins administrativos, a interface do GeoNode estende a interface de administração do Django, ou seja, permite reutilizar a interface existente sem a necessidade de separar a vertente administrativa do servidor e do catálogo [27]. Por outro lado, a interface de gestão dos dados permite explorar, processar, estilizar e compartilhar mapas, tal como, dados geoespaciais [28].

Acompanhar tecnologias como o Django e o GeoServer permitem que os catálogos sejam facilmente escaláveis e com grandes níveis de performance uma vez que aproveita recursos destas tecnologias.

	GeoNetwork	GeoNode
Open-source	✓	✓
Padrões da <i>OGC</i>	✓	✓
Features de renderização	✓	✓
Features de filtragem	✓	✓
Exportar/Importar informação	✓	✓
Integração com o GeoServer	✓	✓
Integração com Python	✓	✓
Escalabilidade	×	✓

Tabela 2.3: Comparação entre as aplicações de catálogo - Baseado na informação retirada de [26] [29]

A escolha de uma aplicação de catálogo é difícil visto que ambas as tecnologias são bastante semelhantes e cobrem os propósitos de utilização. É uma questão que deve ser analisada com mais detalhe no futuro para selecionar a tecnologia mais adequada para o sistema FireLoc. Contudo, tendo como base a descrição feita de cada uma das tecnologias e a comparação na tabela 2.3, podemos pressupor que o GeoNode será a tecnologia que mais se adequa ao projeto, dada a sua garantia de escalabilidade. Para além disso, a integração com o Python relativamente ao GeoNetwork não é feita através de bibliotecas, mas através de pedidos à API correspondente. Por outro lado, o GeoNode, ao estender a aplicação de administração do Django, é integrado por si só.

Sumário

Na pequena revisão das tecnologias de web mapping salientou-se e justificaram-se as escolhas feitas para o sistema FireLoc. Introduziu-se uma série de conceitos importantes e apresentou-se as tecnologias que constituem o projeto FireLoc, tal como, possíveis tecnologias a integrar nesta vertente (aplicações de catálogo). Explicou-se, também, alguns conceitos que têm sido um critério comum na seleção das tecnologias até o momento, como a interoperabilidade e os padrões OGC.

2.5.2 Tecnologias em Front-end

As tecnologias de front-end, apesar de não influenciarem a lógica da aplicação, têm de ser escolhidas de forma a que a UI seja compatível em qualquer dispositivo móvel e também na web.

O FireLoc pretende ter 2 aplicações no seu sistema:

- Uma aplicação web, que será o portal da aplicação que integra as tecnologias de renderização de mapas, o Leaflet, e que permite a análise e visualização de detalhes dos incêndios identificados pelo sistema, assim como, as contribuições, entre outros;
- Uma aplicação móvel, que é a principal fonte de dados, e que permite a identificação e processamento da informação dos utilizadores para denunciar a existência de incêndios florestais. A informação proveniente desta aplicação é processada pelos módulos independentes que integram o sistema FireLoc.

Tendo em conta as necessidades do sistema identificadas acima, surgem 3 conceitos importantes: aplicações móveis nativas, aplicações web, e aplicações móveis híbridas.

Aplicações nativas

As aplicações nativas são muito úteis e extremamente eficientes para desenvolvimento de aplicações móveis pois são, como o próprio nome indica, nativas ao dispositivo móvel. Isto permite fácil acesso a funcionalidades singulares e recursos nativos dos dispositivos como a câmara, o GPS, o acelerômetro, a bússola, entre outros. Para o FireLoc é essencial o acesso a estes recursos, principalmente do GPS, para determinar a localização dos utilizadores e, da bússola, para determinar a orientação relativamente ao norte. São funcionalidades que devolvem dados importantes para o sistema utilizar na identificação do incêndio florestal [39].

Aplicações web

Apesar de não terem acesso a funcionalidades e recursos nativos, como uma aplicação móvel nativa o permite, as aplicações web podem correr em dispositivos móveis. São caracterizadas por facilitarem o processo de desenvolvimento, uma vez que, utilizam linguagens comuns da web, como HTML, CSS e JavaScript. [39].

As aplicações web resolvem o problema do custo de desenvolvimento da aplicação mas é desfavorável pela limitação estrita da utilização de recursos nativos dos smartphones.

Aplicações híbridas

As aplicações híbridas permitem conciliar as limitações das duas abordagens previamente descritas, e aproveitar as vantagens de ambas. A grande vantagem das aplicações híbridas é permitir o desenvolvimento com baixo custo pela adoção de tecnologias da web por garantirem portabilidade em qualquer smartphone. Permitem, também, ter acesso a alguns recursos nativos dos dispositivos que serão úteis para o sistema [39].

Desta forma, é necessário analisar as vantagens e desvantagens de cada uma das abordagens e decidir qual é o tipo de aplicação que se adequa mais ao projeto do FireLoc.

	Aplicações Nativas	Aplicações Web	Aplicações Híbridas				
Custo de Desenvolvimento				<table border="1"> <tr> <td style="background-color: #FFC0CB;">Desvantagem</td> </tr> <tr> <td style="background-color: #A9A9A9;">Neutral</td> </tr> <tr> <td style="background-color: #C0FFC0;">Vantagem</td> </tr> </table>	Desvantagem	Neutral	Vantagem
Desvantagem							
Neutral							
Vantagem							
Portabilidade							
Acesso a recursos nativos							
Distribuição							
Segurança dos dados							
Performance							

Tabela 2.4: Comparação entre aplicações: Nativa, Web e Híbrida [72]

Com base na tabela 2.4, as aplicações nativas são, sem dúvida, as que têm mais qualidade em relação ao acesso dos recursos do dispositivo e garantias segurança. Contudo, a variedade de smartphones que existem tornam muito difícil desenvolver uma aplicação que se adapte a qualquer dispositivo móvel, tornando o seu desenvolvimento bastante complexo e com um custo elevado. Por outro lado, as aplicações web acabam por ser excluídas do desenvolvimento da aplicação móvel do FireLoc uma vez que não tem qualquer forma de aceder aos recursos do dispositivo. Por fim, as aplicações híbridas como já referido, permitem ter o melhor das duas abordagens e deverá ser a escolhida, principalmente por não ter desvantagens na comparação entre as abordagens descritas [10]. Para o desenvolvimento do portal torna-se evidente a utilização de uma abordagem totalmente web.

Angular

Para complementar a decisão, deverá ser escolhido um framework para o desenvolvimento do portal (web) e da aplicação móvel (híbrida). Algumas tecnologias muito conhecidas como React, Angular e Vue, são candidatos indiscutíveis. O FireLoc optou por escolher Angular para o desenvolvimento das suas aplicações devido à familiaridade da equipa em relação a esta tecnologia, o que facilita muito o desenvolvimento da aplicação. Para o desenvolvimento da aplicação móvel, escolheu-se uma tecnologia complementar que atue em paralelo com o Angular, o Ionic.

Ionic

O Ionic permite-nos, através de bibliotecas externas, ter acesso aos recursos nativos dos smartphones com uma boa performance e um rápido ciclo de vida de desenvolvimento. Apesar de algumas outras tecnologias, como NativeScript (mesma natureza de Ionic), não precisarem de bibliotecas externas e permitirem acesso direto aos recursos nativos dos smartphones, as aplicações desenvolvidas com esta tecnologia tendem a ser muito mais

complexas por não suportar HTML. NativeScript, comparativamente com o Ionic, apresenta mais performance uma vez que não necessita de plugins para acesso aos recursos do smartphone.

Para evitar complicações no desenvolvimento da aplicação móvel e, tendo em conta que Ionic tem uma comunidade muito mais vasta que o NativeScript, é recomendável optar por usar Ionic em paralelo com Angular. O uso das bibliotecas externas não será um problema visto que toda lógica da sistema induz na integração e exploração de tecnologias independentes.

2.5.3 Tecnologias em Back-end

As tecnologias de back-end são muito importantes para um projeto, pois ditam toda a lógica da aplicação. Tipicamente, o back-end tem uma justificação mais forte em termos de escolha em comparação com o front-end.

No caso do FireLoc, selecionaram-se algumas tecnologias de back-end e os frameworks associados às mesmas para as comparar e eleger a mais adequada para o projeto. Os candidatos são Django (Python), NodeJS (JavaScript) e Laravel (PHP) por serem os mais utilizados para o desenvolvimento web e terem na sua infraestrutura interna ferramentas que facilitam o desenvolvimento, como segurança e escalabilidade.

Django

Django é um framework de Python que tem uma arquitetura padronizada e muito conhecida, Model-Template-View (MTV). Esta arquitetura permite estruturar a aplicação de forma organizada e bem definida, tornando o Django uma ótima escolha para projetos de grande escala, como o FireLoc. É extremamente simples a iteração com a base de dados do sistema, por basear-se no Object-relational mapping (ORM). A técnica ORM assume que as tabelas de uma base de dados são representadas através de classes/objetos e os registos de cada tabela são representados como instâncias das classes correspondentes [86], facilitando a manipulação, acesso e aplicação de restrições na base de dados do sistema.

Ainda, como já foi revelado anteriormente, o Django oferece uma interface de administração para a aplicação o que facilita a administração de aplicações de grande escala. Tem um grande destaque quanto ao desenvolvimento de aplicações GIS, uma vez que o processamento e análise de informação geoespacial com Python é simplificado com bibliotecas externas como GDAL, GRASS e GeoPandas [23].

Laravel

Por outro lado, Laravel é um framework de PHP que tem uma arquitetura padronizada e também muito conhecida, Model-View-Controller (MVC). Ao contrário do MTV, os controllers da aplicação gerem toda a lógica de negócio e tendem a renderizar views desenvolvidas no front-end à medida que é necessário. As funcionalidades são dinâmicas e adaptáveis, o que permite evoluir uma aplicação de pequena escala para grande escala sem complexidade. Apesar das suas qualidades, no que toca à comunidade e a integração de bibliotecas externas não supera outros frameworks como Node [23]. O Django e o Laravel tem grandes ferramentas que garantem segurança na aplicação, evitando Cross-site scripting e SQL Injection [23].

NodeJS

O NodeJS é um framework de JavaScript altamente reconhecido para desenvolvimento web uma vez que não obriga a distanciar as tecnologias do front-end e back-end, o que permite uma redução na curva de aprendizagem e de separação de conceitos entre as aplicações (JavaScript based). O NodeJS destaca-se em relação aos outros frameworks pela sua capacidade de gerir os recursos do sistema por se basear na execução do código em single-thread. Contudo, NodeJS permite computação assíncrona o que torna difícil a sua gestão e harmonização. Para além disso, possui o maior gerenciador de pacotes/bibliotecas externas da atualidade, o Node Package Manager (NPM) [23], tendo valor pois permite-nos integrar bibliotecas úteis para processamento de dados geoespaciais e destaca-se em termos de escalabilidade da aplicação em comparação ao Django e Laravel. Com o NodeJS, há um controlo mais estrito da aplicação por responsabilizar os developers na construção de toda a estrutura, incluindo segurança, o que nem sempre é benéfico.

	Django	Laravel	Node
Linguagem	Python	PHP	JavaScript
Adequado para aplicações GIS	✓	×	✓
Segurança	✓	✓	×
Performance	✓	✓	✓
Desenvolvimento de API	✓	✓	✓
Familiaridade	✓	×	✓
Comunidade	✓	×	✓

Tabela 2.5: Comparação entre frameworks de server-side

Analisando a tabela 2.5 e a descrição de cada uma das frameworks, torna-se evidente que, a melhor escolha para este projeto é o Django, uma vez que é adequado e altamente recomendável para aplicações que envolvam tecnologias e dados GIS. Apesar do Node ser adequado para GIS, o Django supera-o por ter bibliotecas dedicadas nesta área e facilita, também, o acesso e gestão da base de dados. O Laravel, para este tipo de aplicações, não seria a melhor opção e prejudicaria o desenvolvimento, pelo que é sensato excluir este framework.

2.5.4 Base de Dados

As bases de dados são essenciais para armazenar qualquer tipo de dados num sistema. Contudo, nem todas as bases de dados existentes têm recursos que permitam armazenar dados geoespaciais. Nesta secção são descritas e comparadas algumas bases de dados com extensões que permitem manipular dados georreferenciados.

Existem inúmeras vantagens na utilização de bases de dados relacionais para aplicações GIS que, em complemento com o GeoServer, modelam o sistema para que a comunicação e os pedidos sejam mais simples e ágeis, isto é, com maior capacidade de devolver a informação pretendida. Neste sentido, selecionaram-se três bases de dados que possibilitam operar com informação geoespacial, MySQL, PostgreSQL e Oracle Spatial.

O MySQL e o PostgreSQL são as duas bases de dados relacionais mais famosas na atualidade [55], ambas têm uma performance acima da média e conseguem ser usadas para projetos com Data Warehouses para análise de Big Data, entre outras finalidades. São muito semelhantes em termos da sua estrutura e suportam uma grande variedade de tipos de dados, em particular os espaciais e geométricos, o que faz delas ótimas candidatas para

a base de dados do FireLoc [11]. Contudo, o PostgreSQL destaca-se em muitas funcionalidades nesta vertente geoespacial.

PostgreSQL

A extensão do PostgreSQL, PostGIS, tem uma documentação surpreendente vasta e aborda todas as operações que a extensão permite, desde a manipulação de objetos vetoriais e raster até operações de transformações e acesso a topologia dos objetos espaciais (arestas, vértices, entre outros). Por outro lado, a documentação do MySQL desilude quando é comparada a do PostGIS, dado que não possui muitas funções aplicáveis nas geometrias, o que limita consideravelmente as operações que se podem realizar sob os dados georreferenciados. Uma das limitações mais evidentes é não suportar rasters nem transformações de um sistema de referência espacial para outro. É vital mencionar que o PostGIS é suportado pelo servidor e gerador de serviços de dados geoespaciais do FireLoc, o GeoServer. O MySQL consegue, também, ser suportado pelo GeoServer, mas com a necessidade de instalação de um plugin externo para habilitar a base de dados [32]. Este ponto é, igualmente, necessário para o Oracle Spatial.

Oracle Spatial

O Oracle Spatial é, também, uma base de dados que permite operar sob dados espaciais vetoriais e, em relação às funcionalidades é muito semelhante ao PostGIS. No entanto, em termos de performance é bastante lento a operar sob os dados. A razão da baixa performance do Oracle é devido ao facto de ter o seu próprio Geometry Engine, isto é, algoritmos exclusivos da Oracle que processam as geometrias. Isto não acontece com o PostGIS, uma vez que utiliza Geometry Engine - Open Source [58]. Uma comparação entre o Oracle Spatial e o PostGIS já foi feita num case study pelo Instituto de Tecnologia da Universidade de Nirma, onde se verificou que esta diferença entre os Engines provocou uma disparidade mínima de 50% e máxima de 450%, aproximadamente, na performance de execução de queries [20].

Feita esta breve descrição de cada uma das bases de dados seleccionadas, depreende-se que o PostgreSQL com extensão espacial (PostGIS) é a base de dados mais adequada para o projeto FireLoc.

2.5.5 Tecnologias de Virtualização

As tecnologias de virtualização são tecnologias que permitem virtualizar aplicações e/ou servidores para que estes possam usar a capacidade total de uma máquina física (Hardware). Desta forma, à medida que forem criados ambientes virtuais, a máquina física distribui os seus recursos e, desta maneira, as aplicações/servidores assumem o uso da máquina.

As principais tecnologias de virtualização incidem no hardware, nomeadamente, o uso de máquinas virtuais e a containerização de software com o uso containers para “empacotar” software e torná-lo completamente independente. Ambas tecnologias são complementares, e, desta forma, permitem tirar o melhor proveito de cada uma [5].

O sistema FireLoc utiliza o Amazon Web Services (AWS) como tecnologia de virtualização de hardware para executar os servidores e aplicações do sistema sem preocupações com a

gestão da escalabilidade e disponibilidade, uma vez que os serviços da AWS garantem estes atributos. O objetivo é que o foco das nossas equipas seja, maioritariamente, no desenvolvimento de código e processamento de informação, e não na gestão da infraestrutura do sistema.

As tecnologias utilizadas no sistema FireLoc são Open Source e independentes, havendo, assim, necessidade de utilizar o Docker como tecnologia de containerização para empacotamento dos servidores e das aplicações utilizadas. O Docker possibilita que se consiga independizar as aplicações e, de certa forma, as poderemos adaptar às necessidades do projeto. O sistema FireLoc utiliza, atualmente, quatro containers independentes:

- Container da API (Servidor);
- Container do PostgreSQL (Base de Dados);
- Container do GeoServer (Componente do SDI);
- Container de Backup PostgreSQL.

Estes containers comunicam entre si, pois estão configurados para tal, no entanto são independentes. No Capítulo 5 será descrito de forma mais detalhada como é que foi realizada a configuração e preparado o ambiente de produção.

2.5.6 Sumário

Como referido inicialmente nesta secção, pretendeu-se clarificar e justificar as escolhas feitas pela equipa FireLoc em relação às tecnologias a usar. O processo de seleção já tinha sido realizado em anos anteriores, pelo que, esta secção, limitou-se a apresentar as tecnologias e comparar, de forma sucinta, as tecnologias atuais com as alternativas existentes.

Na tabela 2.6, encontra-se de forma resumida as tecnologias escolhidas em cada vertente tecnológica, web mapping, front-end, back-end, base de dados e virtualização/containerização.

	Tecnologia
Web-Mapping	<ul style="list-style-type: none"> • Leaflet - Representação de mapas • GeoServer - Criação de Serviços • Protocolos da OGC
Front-end	<ul style="list-style-type: none"> • Angular - Aplicação Web Cliente • Ionic/Angular - Aplicação Móvel Cliente
Back-end	<ul style="list-style-type: none"> • Python - Linguagem de Programação • Django - Framework de Desenvolvimento
Base de Dados	<ul style="list-style-type: none"> • PostgreSQL - Base de Dados • PostGIS - Extensão PostgreSQL
Virtualização	<ul style="list-style-type: none"> • Docker • AWS

Tabela 2.6: Tecnologias escolhidas por vertente tecnológica

Capítulo 3

Especificação de Drivers Arquiteturais

O objetivo da especificação dos drivers arquiteturais é definir e detalhar os requisitos associados ao sistema FireLoc, assim como os atributos de qualidade. Os requisitos funcionais são drivers que descrevem o comportamento funcional do sistema numa série de condições específicas, e que representam as necessidades do ponto de vista funcional do sistema e, na prática, traduzem-se nas funcionalidades que estão à disposição dos utilizadores na versão final do projeto. Geralmente, vêm acompanhados com critérios de aceitação/sucesso, de forma a que se consigam testar e avaliar as funcionalidades do sistema.

A avaliação dos requisitos é muito importante para a validação dos mesmos e fechar o ciclo a que estes estão sujeitos: Elicitação \rightarrow Análise \rightarrow especificação \rightarrow Validação. Uma vez que se fecha o ciclo de um requisito, podemos assumir que a sua implementação está completa.

Por outro lado, os atributos de qualidade são bastante importantes e devem ser especificados, de forma a que seja possível utilizar métricas explícitas para verificar e validar a existência dos mesmos no sistema. Será apresentada uma breve descrição de cada um, seguida de cenários do requisito específico do atributo de qualidade que, na prática, são cenários em que o atributo é notório.

3.1 Requisitos Funcionais - Portal FireLoc

Nesta secção, pretende-se especificar quais são as funcionalidades associadas ao Portal FireLoc, onde serão listados e descritos os requisitos associados às funcionalidades de back-office, tal como, apesar de numa menor quantidade, funcionalidades de front-office. As funcionalidades de back-office são, por exemplo, funcionalidades relacionadas com a gestão de informação, seja esta geospacial ou de outra natureza (utilizadores, grupos, entre outros). As funcionalidades de front-office, por outro lado, estão relacionadas com o consumo de serviços do sistema do lado do cliente.

O Portal encontra-se numa fase inicial de desenvolvimento pelo que, a maior parte dos seus requisitos, apesar de já estarem elicitados e analisados (duas primeiras fases do ciclo de vida), será necessário especificá-los e validá-los. A fase de validação será abordada mais em frente, mais especificamente no Capítulo 6.

Alguns módulos já se encontravam desenvolvidos e passaram por um processo de integração

com as aplicações de cliente, com o objetivo de preparar o sistema para uma exposição pública e apta para uso. Porém, estes módulos sofreram alterações significativas e, por conseguinte, pode-se assumir que não se fechou o ciclo de vida.

Entre esses módulos, podemos destacar o sistema de autenticação, a gestão de utilizadores, a gestão de contribuições, parte da gestão de eventos, e a gestão da grelha de referência geográfica. A fase de integração identificou vários pontos a melhorar nos módulos, pelo que, houve a necessidade de iniciar um novo ciclo para estes requisitos.

O objetivo prático desta dissertação tem como cerne o desenvolvimento da API do sistema FireLoc que, em certos módulos, inclui a gestão de dados geospaciais. Posto isto, são listados todos os módulos (requisitos) em que foi dedicado maior parte do trabalho, e onde será realizada uma descrição e propósito de cada um dos módulos, assim como os componentes que integra.

A especificação dos requisitos são apresentados com Use Case (UC), organizados e agrupados por categoria funcional. A listagem que se segue inclui uma descrição de cada Requisito Funcional (RF). A relação entre o utilizador-funcionalidade será representada no Capítulo 4 na Arquitetura do Sistema. A representação do diagrama de UC geral será omitido pela sua dimensão. Contudo, os diagramas de cada UC podem ser encontrados no Apêndice A apresentando parte da constituição do diagrama geral de UC.

Os UCs que se seguem dizem respeito aos três níveis de abstração mais elevados, excluindo o nível Cloud (mais abstrato de todos) uma vez que esclarecem a lógica da aplicação e do sistema FireLoc. Os três níveis de abstração considerados são Kite, Sea e Fish. Os níveis são metafóricos, onde Kite (nível mais alto após Cloud) diz respeito às categorias funcionais, como, por exemplo, gerir as contribuições dos utilizadores. Descendo o nível, o nível Sea pode ser descrito como objetivos diretos do utilizador e que, na prática, são atividades de negócio e âmbito, como listar ou excluir contribuições do sistema. Por fim, temos o nível Fish que apresenta tarefas secundárias do utilizador para alcançar os objetivos principais, como, por exemplo, filtrar a listagem das contribuições por localização geográfica.



Figura 3.1: Níveis de abstração: UC

Para simplificar a definição de cada um dos UCs, são apresentadas apenas as tabelas de UCs com nível de abstração Sea. As tabelas podem ser encontradas no Apêndice B. Cabe destacar que estas tabelas especificam a garantia mínima e a garantia de sucesso do UC e que, por sua vez, servem de base para definir os critérios de aceitação. No entanto, os critérios de aceitação não se podem basear, apenas, nestas especificações pelo que o tema será abordado mais em frente para clarificar como se especificaram os critérios.

3.1.1 UC - 1: Autenticar

A autenticação no sistema é um requisito típico e essencial para qualquer aplicação, pois todas as funcionalidades estarão dependentes do mesmo. Cada tipo de utilizador terá permissões diferentes, o que implica que sistema de autenticação deve reconhecer as credenciais do utilizador e permitir o acesso à informação que lhe compete. Esta funcionalidade só estará disponível para utilizadores não autenticados, sendo ela comum para as duas aplicações do FireLoc, web e móvel.

UC - 1.1: Login

O sistema deverá permitir ao utilizador aceder às funcionalidades da aplicação após as suas credenciais (e-mail e password) serem validadas. O sistema deverá exibir uma mensagem de erro, caso as credenciais do utilizador estejam inválidas ou alguma falha no processo de autenticação ocorrer.

UC - 1.2: Registrar

O sistema deverá permitir ao utilizador realizar o seu registo na aplicação. O sistema deverá exibir uma mensagem de erro, caso o registo não esteja correto (dados inválidos/campos em falta) ou alguma falha no processo de registo ocorrer. O registo na aplicação implica que seja enviado um email para o utilizador confirmar o pedido do seu registo.

UC - 1.3: Recuperar Password

O sistema deverá permitir ao utilizador recuperar a sua password. A recuperação implica que seja enviado um email para o utilizador confirmar o pedido de recuperação de password. O sistema deverá possibilitar a alteração da password, uma vez esta seja confirmada. O sistema deverá exibir uma mensagem de erro, caso alguma falha ocorra no processo de recuperação de password.

UC - 1.4: Confirmar Registo

O sistema deverá reconhecer o processo de registo e permitir ao utilizador autenticar-se no sistema.

UC - 1.5: Confirmar Recuperação de password

O sistema deverá reconhecer o processo de recuperação de password e permitir ao utilizador recuperar as suas credenciais.

3.1.2 UC - 2: Gerir Utilizadores

A gestão de utilizadores é um requisito que permite controlar os utilizadores e como acedem ao sistema. O acesso às diferentes funcionalidades do sistema será determinado com a atribuição de grupos a determinados clientes. Esta funcionalidade estará disponível apenas para os gestores do FireLoc, o que significa que será este perfil de utilizador o responsável pela gestão dos utilizadores. Este RF é satisfeito através das típicas operações CRUD.

UC - 2.1: Adicionar um Novo Utilizador

O sistema deverá permitir adicionar um novo utilizador. Esta funcionalidade não deve ser confundida com o UC 1.2, uma vez que, apesar de ambas registarem utilizadores, têm propósitos diferentes.

O UC 1.2 limita-se a registar utilizadores regulares, isto é, sem permissões especiais. Por outro lado, o UC 2.1 regista utilizadores não regulares, ou seja, utilizadores com permissões especiais. Estes, por sua vez, têm acesso a mais funcionalidades do sistema. O sistema deverá exibir uma mensagem de erro, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 2.2: Atribuir Grupo a Utilizador

O sistema deverá permitir a atribuição de grupos (perfis) ao utilizador. A atribuição do grupo irá delinear as permissões do cliente no sistema. Esta atribuição não é feita sem deferimento, isto significa que é necessária a aprovação do utilizador. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra. Cabe destacar que, para utilizadores regulares, será atribuído um grupo que lhe conceda acesso limitado. Por outro lado, existem grupos próprios para gestores e administradores, o que lhes possibilita aceder a mais funcionalidades.

UC - 2.3: Atualizar Utilizador

O sistema deverá permitir atualizar a informação dos utilizadores. A atualização da sua informação deverá refletir-se na interação com o sistema. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 2.4: Excluir Utilizador

O sistema deverá permitir excluir utilizadores. A exclusão de um utilizador irá, consequentemente, privá-lo do acesso ao sistema. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 2.5 e 2.6: Listar Utilizadores; e Obter um Utilizador Específico

O sistema deverá permitir a listagem de todos os utilizadores do FireLoc, assim como obter a informação de um utilizador em específico, caso seja solicitado pelo gestor. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.3 UC - 3: Gerir Grupos

A gestão de grupos já foi brevemente abordada na descrição do UC 2.2. O FireLoc atualmente tem 4 grupos de utilizadores: FireLoc (Administrador FireLoc), Risk Manager (Gestor FireLoc, por exemplo, a PCP), Volunteer (Voluntário) e Just a User (Utilizador sem contribuições realizadas).

Cada um dos grupos tem diferentes tipos de permissões no sistema, o que delimita as funcionalidades que cada utilizador pode aceder ou a informação que pode observar. Para ficar mais claro, o grupo FireLoc diz respeito aos administradores do sistema e têm, naturalmente, acesso às funcionalidades de back-office, como a gestão de utilizadores e grupos, entre outros. O grupo Risk Manager é destinado às entidades com experiência em GIS para a gestão e prevenção dos incêndios como, por exemplo, a PCP e o ICNF. O grupo

Volunteer são, como o próprio nome indica, os voluntários do sistema FireLoc, isto é, os utilizadores que contribuem voluntariamente para a denúncia de eventos. O grupo Just a User são utilizadores que ainda não realizaram nenhuma contribuição.

A gestão de grupos tem um papel importante na infraestrutura do sistema FireLoc, sendo vital que apenas os gestores do FireLoc e administradores tenham acesso a determinada informação e a um conjunto específico de funcionalidades, para que a gestão e prevenção de incêndios tenha o sucesso pretendido. Voluntários terão, igualmente, um papel importante mas, sem a necessidade de utilizar funcionalidades de gestão do sistema propriamente dito. A gestão de grupos é satisfeita através das típicas operações CRUD.

UC - 3.1: Listar os Grupos de Utilizador

O sistema deve permitir listar todos os grupos e os utilizadores associados. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 3.2: Adicionar um Novo Grupo

O sistema deve permitir adicionar um novo grupo que poderá ter um conjunto de permissões associadas. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 3.3: Atualizar Grupo

O sistema deverá permitir atualizar os grupos do sistema, como, por exemplo, a associação de permissões. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 3.4: Excluir Grupo

O sistema deverá permitir excluir grupos do sistema. Caso o grupo seja excluído e tiver utilizadores associados, o grupo é, automaticamente, associado ao grupo Just a User. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.4 UC - 4: Gerir Contribuições

A gestão de contribuições é constituída por funcionalidades que permitem controlar as denúncias de eventos (ocorrências de incêndio) por parte dos utilizadores, as suas contribuições. Nem todas as contribuições são válidas, estas têm de ser processadas e validadas por um módulo específico do FireLoc encarregue para o efeito. À medida que o número de contribuições aumenta, a sua gestão torna-se cada vez mais complexa. Desta forma existem funcionalidades de filtragem de contribuições o que facilita a análise e a gestão das contribuições e, conseqüentemente, dos incêndios florestais.

UC - 4.1: Listar contribuições Voluntárias

O sistema deve permitir listar todas as contribuições do sistema. A listagem poderá ser filtrada por: Janela Temporal, Localização geográfica e por Utilizador. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 4.2: Representar Contribuições Voluntárias

O sistema deve permitir representar todas as contribuições do sistema no mapa disponibilizado. A representação poderá ser filtrada por: Janela Temporal, Localização geográfica e por Utilizador. O sistema não deverá representar as contribuições caso o pedido não consiga ser processado ou alguma outra falha ocorra, informando o utilizador sobre a ocorrência.

UC - 4.3: Obter uma Contribuição Específica

O sistema deverá permitir obter uma contribuição específica. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 4.4: Excluir Contribuições

O sistema deverá permitir excluir contribuições do sistema. Ao excluir uma contribuição do sistema, a mesma deverá ser excluída do mapa. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 4.5: Adicionar Novas Contribuições

O sistema deverá permitir adicionar novas contribuições (dos voluntários) ao sistema. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.5 UC - 5: Gerir Eventos

A gestão de eventos é um dos principais módulos exclusivos do FireLoc, desenvolvido para auxiliar na gestão de incêndios.

Os eventos, podem ser contribuições validadas pelo sistema, sendo elas ocorrências de incêndios identificadas pelo sistema FireLoc após uma validação efetuada pelos nossos algoritmos.

Dito isto, é importante destacar que, existem dois tipos de eventos, os eventos identificados pelo sistema e eventos reais. Os eventos reais são ocorrências de incêndios que foram observadas e na qual os seus dados são especificados à priori. Os eventos reais são muito úteis para o sistema FireLoc conseguir, efetivamente, validar se os eventos identificados pelo sistema estão corretos. São usados como um fator de comparação e validação dos eventos identificados pelo sistema. Na secção 2.4 é apresentado um exemplo concreto desta funcionalidade. Neste módulo são apenas abordados os eventos reais, sendo os eventos do sistema discutidos no UC 8 e UC 11.

UC - 5.1: Gestão de Eventos Reais (Observados à priori)

O sistema deve permitir gerir os eventos reais, isto é, os eventos que são observados à priori. A gestão dos eventos reais cobrem as seguintes funcionalidades:

1. Listar todos os eventos reais observados;
2. Obter um evento real específico do sistema;
3. Adicionar um novo evento real observado;
4. Atualizar um evento real observado;
5. Excluir todos os eventos reais observados;
6. Excluir um evento real específico do sistema.

O sistema não deverá realizar nenhuma operação relacionada com os eventos reais (observados) caso o pedido não consiga ser processado ou alguma outra falha ocorra, informando o utilizador sobre a ocorrência.

UC - 5.2: Gestão Temporal dos Eventos

O sistema deve permitir gerir os anos no qual ocorreram eventos reais ou do sistema. A gestão dos anos em relação aos eventos cobrem as seguintes funcionalidades:

1. Listar todos os anos do sistema ou um em particular;
2. Obter um ano específico do sistema;
3. Adicionar um novo ano ao sistema;
4. Atualizar um ano do sistema;
5. Excluir todos os anos do sistema;
6. Excluir um ano específico do sistema.

O sistema não deverá realizar nenhuma operação relacionada com os anos dos eventos caso o pedido não consiga ser processado ou alguma outra falha ocorra, informando o utilizador sobre a ocorrência.

UC - 5.3: Gestão de Áreas Ardidadas

O sistema deve permitir gerir as áreas ardidadas identificadas. A gestão das áreas ardidadas cobrem as seguintes funcionalidades:

1. Listar todas as áreas ardidadas do sistema;
2. Obter uma área ardidada específica do sistema;
3. Adicionar uma nova área ardidada ao sistema;

4. Atualizar uma área ardida do sistema;
5. Excluir todas as áreas ardidadas do sistema;
6. Excluir uma área ardida específica do sistema.

O sistema não deverá realizar nenhuma operação relacionada com as áreas ardidadas do sistema caso o pedido não consiga ser processado ou alguma outra falha ocorra, informando o utilizador sobre a ocorrência.

Este UC não deve ser confundido com o UC 5.1. Apesar de ambos serem eventos reais, em sentido estrito da palavra, um diz respeito ao incêndio propriamente dito e o outro à área que ardeu como consequência desse incêndio.

3.1.6 UC - 6: Gerir Camadas

A gestão de camadas diz respeito às camadas de informação geospacial. As camadas (layers) de um mapa servem para definir como um conjunto de dados geospaciais são representados no mapa e o seu valor simbólico, como as suas propriedades, incluindo as respetivas legendas (labels).

Gerir camadas, por si só, é uma das principais maneiras de trabalhar com dados geospaciais, pelo que é uma funcionalidade essencial para o projeto e que está diretamente relacionada com o GeoServer.

Como foi descrito na secção 2.5.1, mais concretamente no tópico das tecnologias para gerar serviços web-mapping, nomeadamente com o GeoServer, é descrita, de forma sucinta, como que é feita a gestão da informação geospacial neste servidor. Resumidamente, as camadas terão associadas uma série de conceitos do GeoServer como Workspaces, Stores e a camada (layer).

Desta forma, a ideia é que a gestão de camadas na API do sistema FireLoc seja, na verdade, a gestão das referências às camadas que se encontram no GeoServer, desta forma, o acesso ao servidor geospacial (GeoServer) é isolado, e toda a gestão é realizada a partir da API.

UC - 6.1: Adicionar uma Nova Camada Geoespacial

O sistema deverá permitir adicionar camadas geospaciais ao sistema. Adicionar uma nova camada irá ter efeito no GeoServer, isto é, deverá haver um serviço com as especificações da camada adicionada. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.2: Atualizar uma Camada Geoespacial

O sistema deverá permitir atualizar camadas geospaciais do sistema. A atualização de uma camada irá afetar o serviço do GeoServer e, conseqüentemente, a representação no mapa. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.3: Excluir uma Camada Geoespacial

O sistema deverá permitir excluir camadas geoespaciais do sistema. Excluir uma camada implica que o serviço do GeoServer relacionado com a camada seja, também, excluído. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.4: Listar camadas geoespaciais

O sistema deverá permitir listar as camadas geoespaciais do sistema, ou seja, as referências dos serviços do GeoServer existentes. A listagem deve permitir identificar as especificidades do serviço. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.5: Obter uma Camada Geoespacial Específica

O sistema deverá permitir obter uma camada geoespacial específica do sistema. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.6: Atribuir Permissões de Visualização

O sistema deverá permitir atribuir permissões de visualização num determinado grupo de camadas (layer group). Dependendo das permissões as camadas poderão ou não estar visíveis para o utilizador. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.7: Gerir Estilos

O sistema deverá permitir gerir os estilos de cada camada ou grupo de camadas.

1. Listar as propriedades de estilo das camadas;
2. Listar propriedades de estilo de uma camada específica;
3. Adicionar um novo estilo;
4. Atualizar um estilo;
5. Excluir estilo.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 6.8: Gerir Legendas

O sistema deverá permitir gerir as legendas das camadas geoespaciais.

1. Obter todas as legendas e as suas propriedades, acompanhadas da camada associada;
2. Obter todas as legendas e as suas propriedades de uma camada específica;
3. Associar uma nova legenda a uma camada;
4. Atualizar legenda de uma camada em específico;
5. Excluir legendas.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.7 UC - 7: Gerir Gráficos

A gestão de gráficos diz respeito aos gráficos em contexto geospacial. Estes gráficos têm um papel importante para análise de informação pois permitem perquirir o comportamento e progresso dos incêndios, e em que condições acontecem de forma representativa e organizada. Desta forma, este requisito serve como complemento da análise que é feita através da representação dos dados geospaciais nos mapas.

O sistema FireLoc deve estar preparado para receber quaisquer tipos de dados geospaciais e manipular a informação de forma a que possa ser representada em diferentes tipos de gráficos (linhas, barras, scatter, entre outros).

UC - 7.1: Adicionar um Novo Gráfico Geoespacial

O sistema deverá permitir criar um novo gráfico geospacial. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 7.2: Atualizar um Gráfico Geoespacial

O sistema deverá permitir atualizar um gráfico geospacial. A atualização do gráfico deverá refletir-se na informação contida no mesmo (atualizar valores). O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 7.3: Excluir um Gráfico Geoespacial

O sistema deverá permitir excluir um gráfico geospacial. Excluir um gráfico não irá afetar a informação que foi associada ao mesmo. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 7.4: Atribuir Requisitos de Permissões de Visualização

O sistema deverá permitir atribuir requisitos de permissões a um gráfico geospacial, limitando o acesso à visualização de um gráfico, dependendo das permissões atribuídas a cada grupo de utilizadores. O sistema deverá exibir uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 7.5: Gerir Dados

O sistema deverá permitir gerir os dados de um gráfico geospacial.

1. Obter os dados de um gráfico geospacial;
2. Associar dados a um determinado gráfico geospacial;
3. Atualizar os dados de um gráfico geospacial;
4. Excluir dados de um gráfico.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.8 UC - 8: Gerir Resultados do Processo de Localização de Eventos

A gestão dos resultados do processo de localização de eventos (incêndios) é uma das funcionalidades com mais componentes e está relacionado com os eventos identificados pelo sistema FireLoc. Este módulo, tem quatro categorias distintas, cada uma sub-dividida nos componentes que lhe correspondem. Das quatro categorias podemos destacar:

1. Atributos da Localização de Incêndios;
2. Resultados da Avaliação da Localização de Incêndios;
3. Abordagens da Localização de Incêndios;
4. Procedimentos da Localização de Incêndios.

UC - 8.1: Gerir Atributos da Avaliação

O sistema deverá permitir gerir os atributos dos resultados do processo de localização de eventos. A gestão destes atributos incide nas típicas operações CRUD incluindo os valores dos atributos.

1. Listar os atributos da avaliação;
2. Adicionar atributo;
3. Atualizar atributo;
4. Excluir atributo.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 8.2: Gerir Resultados da Avaliação

O sistema deve permitir gerir os resultados da avaliação do processo localização de eventos. A gestão dos resultados da avaliação é um componente que permite identificar as avaliações que detetaram incêndio ou não. Os resultados estão associados diretamente com uma abordagem (UC 8.3) e com um procedimento (UC 8.4).

1. Listar os resultados da avaliação;
2. Adicionar resultado;
3. Atualizar resultado;
4. Excluir resultado.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 8.3: Gerir Abordagens da Avaliação

O sistema deve permitir gerir abordagens de avaliação. A Gestão de Abordagens da Avaliação é um componente que permite identificar a abordagem de avaliação utilizada. Neste contexto, uma abordagem pode ser descrita como a técnica utilizada para a identificação de incêndios, como, por exemplo, Métodos Probabilísticos, Métodos de Apoio a Decisão, Machine Learning, entre outros. É importante controlar as abordagens de uma avaliação pois, para cada abordagem utilizada, teremos resultados de avaliação diferentes.

1. Listar os abordagens da avaliação;
2. Adicionar abordagem;
3. Atualizar abordagem;
4. Excluir abordagem.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 8.4: Gerir Procedimentos

O sistema deve permitir gerir os procedimentos dos resultados do processo de localização de eventos. Este é um módulo que está, como vimos, associado ao UC 8.2. Os procedimentos são constituídos por um conjunto de passos, em que cada um corresponde a execução de um script. Neste contexto, os scripts estão relacionados com a identificação e localização de um incêndio. Um passo só poderá ser executado se o passo que lhe antecede se encontrar concluído. É assumido que o procedimento está concluído quando todos os passos estão concluídos.

1. Listar os procedimentos;
2. Adicionar procedimento;

3. Atualizar procedimento;
4. Excluir procedimento.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.9 UC - 9: Gerir Dados Raster

A gestão de dados raster é, na verdade, um conjunto de componentes interligados. Para a gestão de dados raster ter efeito deve-se ter em consideração uma série de componentes complementares e outros módulos como será descrito de seguida.

UC - 9.1: Gerir Tipos Raster

O sistema deve permitir gerir os tipos raster. Os Raster, como já vimos, podem tomar vários tipos, como, por exemplo, o Modelo Digital de Elevação (2.14). É, por isso, que é necessário ter este tipo de informação no sistema FireLoc e poder geri-la conforme necessidade com as típicas operações CRUD.

1. Listar Tipos Raster;
2. Adicionar Tipo Raster;
3. Excluir Tipo Raster.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 9.2: Gerir Datasets Raster

O sistema deve permitir gerir datasets raster. A gestão de datasets raster serve para gerir as referências das camadas raster. Como referido no Capítulo 2, na secção do GeoServer, este servidor divide o processamento da informação em várias categorias e uma delas é precisamente as camadas. Este módulo pretende fazer um processo semelhante e criar um conjunto de camadas que se traduzem em Datasets para organizar e gerir melhor a informação raster que é armazenada.

1. Listar Datasets e/ou Obter Dataset específico;
2. Adicionar Dataset;
3. Atualizar Dataset;
4. Excluir Datasets /ou Excluir Dataset específico.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 9.3: Gerir Camadas Raster

O sistema deve permitir gerir as camadas raster. A gestão de camadas raster está associada, como acabamos de ver no UC 9.2, a datasets raster.

1. Listar Camadas e/ou Obter Camada específica;
2. Adicionar Camada.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 9.4: Adicionar Ficheiros Raster

O sistema deve permitir adicionar Ficheiros Raster. Adicionar ficheiros raster é um módulo que apenas tem uma operação: guardar os ficheiros raster no sistema de ficheiros.

Guardar estes ficheiros implica que seja feito algum processamento de informação geospacial e associação com outros módulos do sistema, nomeadamente, a gestão da grelha de referência geográfica. A grelha de referência foi referida no início do corrente capítulo, no entanto, será abordada e descrita com mais detalhe no Capítulo 5 para que se consiga clarificar para quê e como foi usada para o desenvolvimento deste componente.

3.1.10 UC - 10: Gerir Informação Meteorológica

A Gestão de Informação Meteorológica diz respeito a todos os dados relacionados com a obtenção de dados meteorológicos que são fornecidos por APIs externas públicas, como, por exemplo, a API do Instituto Português do Mar e da Atmosfera (IPMA), entre outras.

Alguns dos módulos obrigam que seja feito algum processamento de informação geospacial e que estará associada a outros módulos, o que obrigou o desenvolvimento de alguns scripts para automatizar a extração dos dados das APIs. O processo de automatização será abordado e descrito com mais detalhe no Capítulo 5.

UC - 10.1: Gerir Previsões Meteorológicas

O sistema deve permitir gerir as previsões meteorológicas do sistema.

1. Listar previsões e/ou Obter previsão específica;
2. Adicionar previsão;
3. Atualizar previsão;
4. Excluir previsão /ou Excluir previsão específica.

Adicionar novas previsões implica adicionar as suas variáveis (UC 10.5) associadas, como, por exemplo, a velocidade do vento, a sua direção, entre outros. As previsões estão associadas a uma determinada localização, o que implica fazer o processamento de informação geospacial. Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 10.2: Gerir Observações Meteorológicas

O sistema deve permitir gerir as observações meteorológicas do sistema.

1. Listar observações e/ou Obter uma observação específica;
2. Adicionar observação;
3. Atualizar observação;
4. Excluir observação e/ou Excluir uma observação específica.

Adicionar novas observações implica a adição das suas variáveis (UC 10.5) associadas, como, por exemplo, a temperatura, a humidade, entre outros. As observações estão associadas a uma estação o que implica que esta informação seja tida em conta (UC 10.4). Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 10.3: Gerir Fontes Meteorológicas

O sistema deve permitir gerir as fontes meteorológicas do sistema.

1. Listar observações e/ou Obter observação específica;
2. Adicionar observação;
3. Atualizar observação;
4. Excluir observação /ou Excluir observação específica.

As fontes são as APIs propriamente ditas. É importante para o sistema FireLoc conseguir rastrear a origem dos valores das variáveis e, por esse motivo, têm uma fonte meteorológica associada. Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 10.4: Gerir Estações Meteorológicas

O sistema deve permitir gerir as estações meteorológicas do sistema.

1. Listar estações e/ou Obter estação específica,
2. Adicionar estações;
3. Atualizar estações;
4. Excluir estações /ou Excluir estação específica.

As estações estão associadas a uma determinada localização o que implica fazer o processamento de informação geospacial. Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 10.5: Gerir Variáveis Meteorológicas

O sistema deve permitir gerir as variáveis meteorológicas do sistema.

1. Listar variáveis e/ou Obter variável específica;
2. Adicionar variáveis;
3. Atualizar variáveis;
4. Excluir variáveis /ou Excluir variável específica.

Como já se referiu no UC 10.1 e UC 10.2, estas variáveis estão associadas aos dados que são extraídos das APIs públicas, relativamente à informação meteorológica. A gestão das variáveis implica, também, associá-las aos seus respetivos valores para cada previsão e/ou observação. Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.1.11 UC - 11: Gerir Resultados do Processo da Classificação de Fotografias

A gestão dos resultados do processo de classificação de fotografias é muito semelhante à gestão dos resultados do processo de localização de eventos (incêndios). Tal como o UC 8, é uma das funcionalidades com mais componentes e está, também, relacionado com os eventos identificados pelo sistema FireLoc. Este módulo, tem três categorias distintas, cada uma sub-dividida nos componentes que lhe correspondem. Das três categorias podemos destacar:

1. Atributos da Classificação de Fotografias;
2. Resultados da Classificação Fotografias;
3. Procedimentos da Classificação Fotografias.

UC - 11.1: Gerir Atributos da Classificação

O sistema deverá permitir gerir os atributos dos resultados do processo de classificação de fotografias. A gestão destes atributos incide nas típicas operações CRUD incluindo os valores dos atributos.

1. Listar os atributos da classificação;
2. Adicionar atributo;
3. Atualizar atributo;
4. Excluir atributo.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 11.2: Gerir Resultados da Avaliação

O sistema deve permitir gerir os resultados do processo de classificação de fotografias. A gestão dos resultados da classificação é um componente que permite identificar as classificações que detetaram incêndio ou não. Os resultados estão associados diretamente com um procedimento (UC 11.3).

1. Listar os resultados da classificação;
2. Adicionar resultado;
3. Atualizar resultado;
4. Excluir resultado.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

UC - 11.3: Gerir Procedimentos

O sistema deve permitir gerir os procedimentos dos resultados do processo de classificação de fotografias. Este é um módulo que está associado ao UC 11.2. Os procedimentos são constituídos por um conjunto de passos, em que cada um corresponde a execução de um script. Neste contexto, os scripts estão relacionados com a classificação de fotografias de um incêndio. Um passo só poderá ser executado se o passo que lhe antecede se encontra concluído. É assumido que o procedimento está concluído quando todos os passos estão concluídos.

1. Listar os procedimentos;
2. Adicionar procedimento;
3. Atualizar procedimento;
4. Excluir procedimento.

Cada uma destas funcionalidades deverá apresentar uma mensagem, caso o pedido não consiga ser processado ou alguma outra falha ocorra.

3.2 Critérios de Aceitação

Os critérios de aceitação são elementos indispensáveis para este projeto pois têm como objetivo validar a conduta dos RFs. Como vimos, os RFs estão representados com UCs, sendo que, cada um deles tem definido uma garantia mínima e uma garantia de sucesso.

Os critérios de aceitação definem as condições para que uma funcionalidade possa fechar o ciclo de vida do RF. Dito isto, cada RF será acompanhado por um critério de aceitação que valide a funcionalidade em si, baseada nas garantias mínimas e de sucesso dos UC, mas não só.

Para definir os critérios de aceitação não basta indicar que todos os testes devem ser ultrapassados, pois podem haver conflitos entre critérios (cumprir um critério leva à falha de outro). Desta forma, pode fazer sentido ter critérios de rejeição, isto é, um RF ter condições que determinam se a funcionalidade não está pronta para ser utilizada por utilizadores finais. Dito isto, serão apresentados três critérios de aceitação e um critério de rejeição, de forma a que as funcionalidades sejam validadas corretamente na fase de testes. Cabe destacar que dois dos critérios de aceitação são comuns para todos os RFs e um particular para cada um. Por outro lado, o critério de rejeição é, também, idêntico para todos.

- Critérios de Aceitação
 1. Garantia de Sucesso de cada UC;
 2. Respeitar a estrutura do Modelo (Serializers - Dependerá do Modelo em questão);
 3. Resposta de Sucesso. (Dependerá do Método HTTP em questão).
- Critério de Rejeição
 1. Tempo médio de resposta do efeito ser superior a 1 segundo¹.

Estes critérios serão tidos em conta na fase de testes e abordados, novamente, no Capítulo 6.

3.3 Atributos de Qualidade

Os atributos de qualidade têm grande importância, pois são eles que descrevem como é que o sistema se comporta, não a nível funcional mas, a nível qualitativo.

A avaliação da qualidade de um sistema é bastante complexa e, por esse motivo, os seus atributos devem estar acompanhados de métricas claras e tangíveis, de forma a possibilitar uma avaliação qualitativa objetiva e uniforme.

Segue-se uma lista dos atributos de qualidade mais prioritários do sistema FireLoc, tendo em conta o contexto em que está inserido e os requisitos e normas seguidas.

3.3.1 Interoperabilidade

A interoperabilidade é um atributo de qualidade que tem sido amplamente referido. O sistema FireLoc teve sempre em consideração garantir este atributo por meio de protocolos definidos pela OGC. Desenvolver o sistema recorrendo a estes protocolos garante que a comunicação com outros sistemas seja possível em Open Format, isto é, partilhar informação e processá-la com sucesso.

Na verdade, testar a interoperabilidade do sistema FireLoc será algo trivial tendo em conta as tecnologias que foram selecionadas no Capítulo 2 para desenvolvimento das soluções web mapping.

¹Segundo a referência em [40], 1 segundo é o limite de tempo em que o utilizador não é prejudicado em relação às suas tarefas nem interrompido o seu fluxo de pensamento.

Fonte	Sistema FireLoc
Estímulo	O Sistema FireLoc faz pedidos de informação geoespacial a um servidor externo cujos serviços se baseiam em protocolos da OGC.
Ambiente	Operacional
Artefactos	Sistema Fireloc, Sistema externo, Protocolos da OGC
Resposta	O sistema FireLoc recebe resposta dos pedidos de forma satisfatória.
Medição da resposta	Receber informação geoespacial de um servidor externo cujos serviços se baseiam em protocolos da OGC.

Tabela 3.1: Cenário de Atributo de Qualidade - Interoperabilidade

3.3.2 Performance

A performance num sistema é um atributo mensurável se considerarmos os cenários corretos. Contudo, este atributo, em particular, pode ser sub-dividido em várias categorias, como, por exemplo, Performance considerando Latência, Performance considerando Renderização de dados e Tempo de resposta. Devido ao principal foco deste trabalho, apenas serão consideradas as categorias de Renderização e Tempo de resposta.

Renderização

A renderização, no contexto do projeto, é considerada como o tempo de carregamento dos dados geoespaciais nos mapas disponibilizados para visualização da informação. A renderização acaba, também, por ser uma variação do tempo de resposta. É um fator muito importante tanto a nível visual como operacional, visto que, muita da informação de interesse, por exemplo, para a PCP, é representado em mapas.

Fonte	Proteção Civil
Estímulo	Um gestor FireLoc faz um pedido ao sistema para representação de informação no mapa
Ambiente	Operacional
Artefactos	Sistema Fireloc
Resposta	O sistema FireLoc renderiza a informação pedida no mapa
Medição da Resposta	A informação pedida deve ser renderizada no mapa até um máximo de de 1 segundo. ²

Tabela 3.2: Cenário de Atributo de Qualidade - Performance (Renderização)

Tempo de resposta

Por outro lado, o tempo de resposta é considerado como o tempo em que a API do sistema FireLoc demora a responder aos pedidos. A API deverá estar preparada para suportar grandes quantidades de pedidos de forma abrupta. É esperado que, nestes momentos, a API sofra um sobrecarregamento de pedidos e deverá ter um tempo de resposta adequado sem prejudicar as tarefas e o fluxo de pensamento dos utilizadores.

²Valores retirados de [40]

³Valores retirados de [40]

Fonte	Utilizador geral
Estímulo	Centenas de pedidos são feitos ao sistema.
Ambiente	Sobrecarregado
Artefactos	Sistema Fireloc
Resposta	O sistema responde ao pedido do utilizador.
Medição da Resposta	O tempo de resposta do pedido deverá estar até um máximo de 1 segundo. ³

Tabela 3.3: Cenário de Atributo de Qualidade - Performance (Tempo de Resposta)

No Capítulo 6 será demonstrado como é que se mediu a Performance no que diz respeito a renderização dos dados geospaciais em mapas e ao tempo de resposta dos pedidos à API.

3.3.3 Segurança

A segurança é um atributo de qualidade especial no que toca na sua definição pois, na realidade, segurança é apenas um conceito e não uma propriedade. No contexto do sistema FireLoc e em qualquer sistema de software é muito importante garantir segurança na aplicação. De forma a descrever o conceito de segurança serão abordadas duas propriedades básicas e fundamentais. No entanto, outras propriedades podem ser incluídas, como Não-repúdio e a Auditoria.

Confidencialidade

O objetivo desta propriedade é garantir a privacidade de informações de forma a que não estejam disponíveis para entidades não autorizadas. Esta propriedade é garantida, geralmente, por algoritmos de encriptação e autorização. O objetivo é limitar o acesso aos utilizadores conforme os grupos aos quais estão inseridos (utilizador comum, gestor, administrador).

Fonte	Utilizador comum
Estímulo	Um utilizador comum tenta apagar todos os utilizadores do sistema FireLoc.
Ambiente	Operacional (Autorização)
Artefactos	Sistema Fireloc
Resposta	O sistema FireLoc nega o acesso ao utilizador comum
Medição da Resposta	O sistema deve negar o acesso à informação e funcionalidades dos quais não tenham permissões e impedir que a operação seja realizada.

Tabela 3.4: Cenário de Atributo de Qualidade - Segurança (Confidencialidade)

No Capítulo 6 será demonstrado como é que o sistema reage aos acessos não autorizados. Será também descrito o sistema de autorização utilizado para garantir o atributo.

Integridade

Este atributo pode ser discutido de duas formas, a integridade dos dados e a integridade do sistema.

A integridade dos dados é garantida com mecanismos que conseguem detetar se alterações no conteúdo da informação foram feitas de forma ilegal ou por entidades autorizadas. Caso seja feita de forma ilegal, a integridade é garantida e a informação não é alterada. Para garantirmos a integridade dos dados, o sistema FireLoc baseia-se nas propriedades Atomicidade, Consistência, Isolamento e Durabilidade (ACID) que são por sua vez garantidas, mais uma vez, pelas tecnologias escolhidas, nomeadamente o PostgreSQL [2].

Fonte	Proteção Cível
Estímulo	Um gestor FireLoc faz um pedido ao servidor para obter informação específica
Ambiente	Operacional
Artefactos	Sistema Fireloc
Resposta	O sistema FireLoc devolve a informação original
Medição da Resposta	A informação que é enviada pelo servidor deve ser exatamente a mesma que é recebida pelo utilizador.

Tabela 3.5: Cenário de Atributo de Qualidade - Segurança (Integridade dos dados)

Por outro lado, a integridade do sistema está relacionada com o propósito do seu desenho (arquitetura). Um sistema desenhado para realizar determinadas tarefas consegue, ou deve, manter a integridade, isto é, realizar as suas operações como esperado.

Fonte	Administrador
Estímulo	Um administrador FireLoc adiciona um novo utilizador ao sistema.
Ambiente	Operacional
Artefactos	Sistema Fireloc
Resposta	O sistema FireLoc insere um novo utilizador ao sistema.
Medição da resposta	O sistema deve realizar, com sucesso, a operação pretendida.

Tabela 3.6: Cenário de Atributo de Qualidade - Segurança (Integridade do sistema)

Neste cenário, dá-se o exemplo de uma operação por parte do Administrador. Um caso em que o sistema não conseguiria garantir a integridade seria, por exemplo, o Administrador querer adicionar um novo utilizador e em vez disso ter excluído todos os utilizadores do sistema.

No Capítulo 6 será demonstrada a integridade do sistema a partir de testes à API.

3.3.4 Disponibilidade

Por último, a disponibilidade tem como objetivo assegurar que os sistemas funcionam prontamente e que os serviços não deixem de estar acessíveis por um grande período de tempo na ocorrência de uma falha.

Fonte	Proteção Civil
Estímulo	Um gestor FireLoc acede ao sistema às 00h00 da madrugada
Ambiente	Operacional
Artefactos	Sistema Fireloc
Respostas	O sistema FireLoc deve permitir o acesso ao utilizador
Medição da Resposta	O sistema deverá estar disponível 99% do tempo. O tempo de indisponibilidade de 1% deverá ocorrer por questões de manutenção. A manutenção não deverá ocorrer mais do que 1 vez por semana e deverá ser realizada em períodos de pouca acessibilidade (entre as 01h00 e as 06h00).

Tabela 3.7: Cenário de Atributo de Qualidade - Disponibilidade

Torna-se bastante complexo testar a disponibilidade de um sistema que se encontra ainda em fases de desenvolvimento e em âmbito académico, uma vez que, idealmente, os testes deveriam ser realizados pelo período mínimo de 1 ano para que os resultados sejam mais confiáveis. Contudo, dada a natureza do trabalho, irá medir-se a disponibilidade durante um mês, todos os dias em cada hora do dia. No Capítulo 6 será descrito de forma mais detalhada como é que este atributo foi testado.

3.3.5 Priorização de Atributos de Qualidade

O sistema FireLoc destaca estes atributos de qualidade em termos lógicos e qualitativos, mas ainda existem outros atributos de qualidade focados mais na interação entre o utilizador e UI. Desses podemos destacar, por exemplo, a Usabilidade.

Qualquer sistema deseja ter o maior número de atributos de qualidade, pois traz confiabilidade ao sistema e, quem o utiliza, estará seguro da sua qualidade. Para isso, fez-se uma pequena priorização dos atributos de qualidade. Na priorização utilizou-se a técnica de Brousseau [48]. Esta técnica utiliza uma notação simbólica para comparar os atributos um a um com os restantes.

Atributo	+ ⁴	Interoperabilidade	Performance	Segurança	Disponibilidade
Interoperabilidade	1		←	↑	↑
Performance	0			↑	↑
Segurança	3				←
Disponibilidade	2				

Tabela 3.8: Priorização de Brousseau - Atributos de Qualidade

Na tabela anterior é demonstrado como é que se realizou a priorização dos atributos de qualidade apresentados. Para que fique claro, a comparação é feita em pares. Para cada célula, na interseção de dois atributos, é feita uma pergunta retórica: “Qual dos dois atributos é mais importante para o sistema?”. Esta questão, em alguns casos, torna-se bastante complicada e pode ocasionar desentendimento se o exercício de priorização for realizado por um grande grupo de pessoas.

Para a atribuição da pontuação e simplificação, utilizou-se o símbolo ↑ para assumir que o atributo da coluna é mais prioritário que o da linha corrente. Por outro lado, o símbolo ← assume que o atributo da linha é mais prioritário que o da coluna. Desta forma, garantimos que cada atributo é comparado com os restantes e é feita uma priorização simples e imparcial.

⁴Somatório da pontuação obtida - priorização

Fundamentação da priorização

A tabela 3.8 mostra a prioridade em relação aos atributos de qualidade descritos, sendo a Segurança o atributo mais prioritário e a Performance o menos prioritário.

A segurança é o atributo de qualidade mais prioritário no sistema FireLoc, pois garante que os seus serviços não serão interrompidos ou afetados por qualquer tentativa de ataque, seja ela intencionada ou não. Consequentemente, podemos garantir a confidencialidade da informação sensível para que esta mesma informação, disponibilizada pelo sistema, seja usada apenas para o contexto em que foi desenvolvido e com as intenções pretendidas. Parte deste atributo é garantido com a utilização da API com uma estrutura baseada na autorização de recursos como o OAuth2 como iremos ver no Capítulo 6.

A disponibilidade e interoperabilidade, apesar de não serem os mais prioritários, são atributos essenciais para o projeto FireLoc e que são garantidos com as metodologias e tecnologias selecionadas para o desenvolvimento do sistema. A interoperabilidade, por exemplo, tem sido um dos principais focos, sendo ele garantido com a abordagem de tecnologias que adotem os padrões da OGC. Por outro lado, a disponibilidade irá ser medida através da monitorização da API de modo a que se consiga observar a acessibilidade dos serviços do sistema e, por sua vez, avaliar a disponibilidade do mesmo.

Por fim, encontra-se a performance do sistema. Não devemos assumir que este atributo não é prioritário nem escusado. A performance é um atributo que em sistemas GIS é fundamental para o processamento da informação geoespacial. É garantido, implicitamente, com tecnologias e configurações nos servidores de geração de serviços (GeoServer) e/ou serviços de renderização de mapas (Leaflet) como Tile Caching e GeoWebCache, isto em relação à performance em termos de renderização. Por outro lado, irá ser abordada, também, a performance em relação ao tempo de resposta da API, de modo a garantir que os serviços não interrompem o pensamento dos utilizadores nem o fluxo das suas atividades.

Capítulo 4

Arquitetura do Sistema

Para clarificar o contexto do sistema e interpretar-se as relações e a interação entre os utilizadores/sistemas externos e a aplicação, será apresentada a arquitetura de software do sistema FireLoc com duas perspetivas distintas, uma para compreender o fluxo de partilha e distribuição dos dados no sistema, e outro, subdividido em 3 níveis de abstração, para compreender o contexto do sistema, as aplicações existentes e os módulos que constituem as aplicações.

Para validar a arquitetura será, também, apresentada uma matriz de rastreabilidade (Traceability). A Traceability terá como objetivo identificar a existência de uma relação direta e lógica entre os drivers arquiteturais identificados e os elementos que constituem a arquitetura. A existência de uma relação significa que os requisitos identificados têm um nível de abstração adequado para a generalização arquitetural exibida, não havendo necessidade de redefinir os requisitos ou a arquitetura. A cooperação entre os diferentes componentes do FireLoc irão formalizar e estruturar a lógica funcional do sistema.

Neste Capítulo pretende-se clarificar o funcionamento e a cooperação entre componentes e, por esse motivo, será apresentada apenas uma categoria de visão arquitetural: Visão funcional/lógica. Dentro desta visão irá ser apresentada e descrita a arquitetura 3-Tier (3 camadas: Client, Logic, Data) complementada por Shared Data e C&C (Components and Connectors) e uma arquitetura C4, na que serão apenas descritos os 3 níveis mais altos de abstração (Context, Container, Component).

4.1 Modelo C4 - Context, Container, Components, Code

O motivo por, apenas, se apresentar os três níveis mais altos de abstração do modelo C4 é pelo facto do nível mais baixo (Code) não ser relevante no presente contexto. Não se pretende clarificar a arquitetura MTV usada para a implementação dos componentes e, por essa razão, este nível será omitido.

4.1.1 Diagrama de Contexto do sistema

O diagrama que se segue representa o contexto do sistema FireLoc e como os diferentes perfis de utilizador interagem com o mesmo.

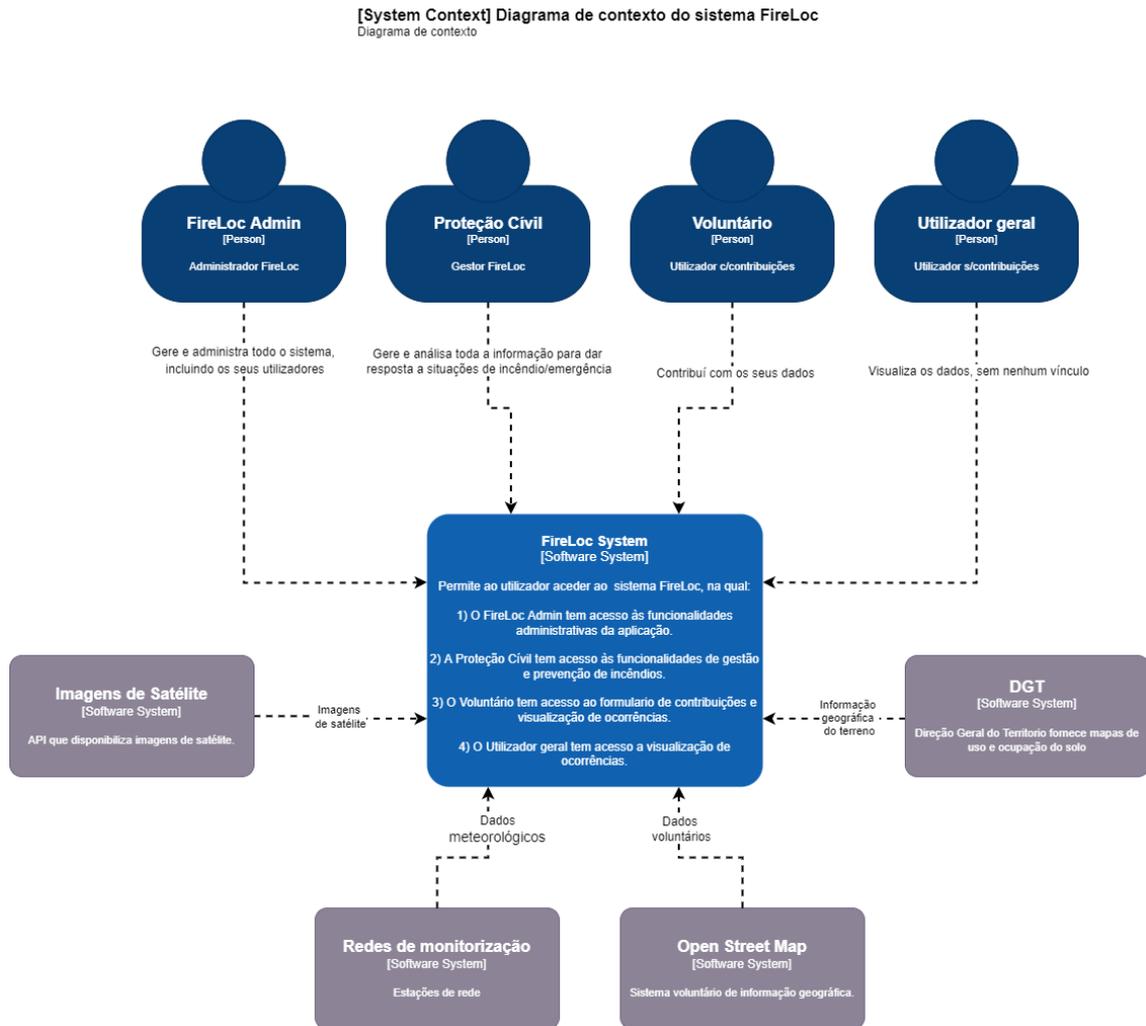


Figura 4.1: Diagrama de Contexto

Como se pode observar, o diagrama de contexto não tem um nível de detalhe demasiado pormenorizado, mas sim um nível capaz de representar os sistemas existentes e os tipos de utilizadores assim como a interação dos mesmos com o sistema. Tem como objetivo representar o nível mais alto de abstração da arquitetura que, por sua vez, se destina para qualquer stakeholder seja ele técnico ou não.

O FireLoc tem neste momento quatro tipos de utilizador representados no topo do diagrama. Estes utilizadores interagem com o sistema FireLoc através das duas aplicações existentes, o Portal e a aplicação móvel. O sistema ainda comunica com entidades externas, neste caso a DGT, um sistema que providencia imagens de satélite com dados úteis para o FireLoc, Redes de monitorização e o OSM.

O serviço disponibilizado pela DGT fornece informação bastante útil (linhas de água (rios), estradas, montanhas, entre outros) e complementa os serviços que o FireLoc pretende oferecer. Por outro lado, as Redes de Monitorização fornecem informação meteorológica como observações e previsões. O OSM fornece dados voluntários de informação geográfica e que são, igualmente, úteis para o sistema, como pontos de interesse (escolas, hospitais, entre outros). Cabe destacar que, apesar do OSM ser um dos sistemas que podem providenciar estes dados, existem outros, como o Foursquare, Facebook e Flickr, que nos permitem identificar os pontos de interesse. Por fim, as imagens de satélite deverão ser fornecidas pela SentinelHub, uma API apoiada pela União Europeia para disponibilizar imagens de

satélite em Open Format, por meio de protocolos da OGC ou por serviços REST.

4.1.2 Diagrama de Container

No nível dois de abstração é representado o diagrama de container do sistema. É explorado o Sistema de Software FireLoc para aumentar a especificidade e detalhe do mesmo.

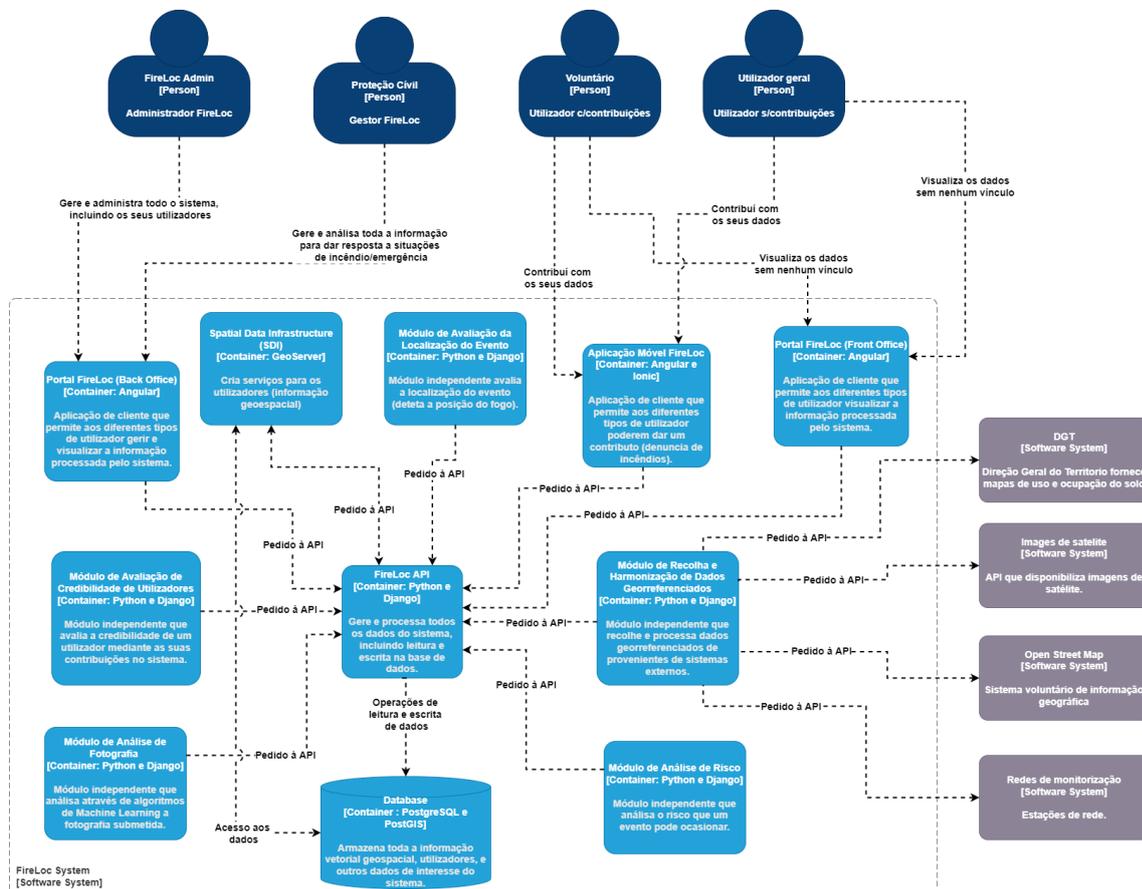


Figura 4.2: Diagrama de Container

No diagrama estão representadas as aplicações e módulos independentes que, no seu conjunto, compõe o sistema FireLoc. Os containers representam, desta forma, os módulos do sistema sejam eles de execução e processamento, como também, de armazenamento.

Neste caso em particular, podemos distinguir duas aplicações, o Portal e a Aplicação Móvel, assim como a Base de Dados do sistema e ainda os Módulos de processamento e análise de informação. Todos estes containers comunicam com um container em especial, a API FireLoc, que controla todas as operações de leitura e escrita de informação.

O Portal já foi mencionado várias vezes como sendo a aplicação de back-office, permitindo gerir e administrar os dados e os utilizadores. Por esse motivo, os tipos de utilizador FireLoc Admin e a PCP terão principal interesse nesta aplicação. Por outro lado, os utilizadores voluntários ou gerais terão interesse em utilizar a aplicação móvel. Apesar do Portal ser a aplicação para back office, tem uma versão para utilizadores voluntários e gerais (front-office) no qual podem visualizar a informação processada pelo sistema, as suas contribuições, entre outras informações.

Ambas as aplicações interagem diretamente com a API que, dependendo do pedido reali-

zado pelas aplicações, irá recorrer aos componentes da API.

O SDI é constituído principalmente pelo GeoServer, sendo um módulo independente que irá comunicar com a API para criar serviços de informação geoespacial. Uma vez criados estes serviços podem ser consumidos pelo cliente (Leaflet) e, conseqüentemente, responder aos pedidos do utilizador.

O módulo de Avaliação de Credibilidade de Utilizadores é, também, independente, e é invocado para avaliar a credibilidade de um utilizador baseando-se na informação fornecida pelo mesmo. Este módulo tem como principal objetivo relacionar os acontecimentos com o utilizador e identificar se as suas contribuições são suspeitas, isto é, identificar indivíduos piromaníacos. A probabilidade de um utilizador contribuir muitas vezes é bastante reduzida e detetar atividade constante de um utilizador pode ser questionável.

A Análise de Fotografias é um módulo que utiliza modelos de Machine Learning para a classificação de incêndios florestais mediante as fotografias disponibilizadas pelos utilizadores, o que permite reduzir o número de falsas contribuições.

O módulo de Avaliação de Localização do Evento é importante para determinar a posição dos utilizadores e, também, do incêndio mediante as informações por eles submetidas nas suas contribuições.

A Recolha e a Harmonização (processamento) de Dados Georreferenciados é o container responsável por obter todos os dados de interesse para o sistema FireLoc provenientes de sistemas ou APIs externas. Para tal, irá comunicar diretamente com as aplicações externas descritas no Diagrama de Contexto.

Por último, o módulo de análise de risco irá determinar, mediante a informação disponível no sistema, o risco de uma zona de modo a que os utilizadores possam ter as devidas precauções.

Com esta pequena descrição pode-se, então, assumir que a API FireLoc tem um papel fundamental, pois controla toda lógica funcional do sistema, merecendo, assim, uma análise mais detalhada como veremos no diagrama de componentes.

4.1.3 Diagrama de Componentes

Aumentando o nível de abstração e focando no principal container do sistema, a API FireLoc, é representado o diagrama de componentes.

Este último diagrama do modelo permite compreender como é que os módulos interagem com a API. Alguns módulos podem interagir com a base de dados sem necessariamente passarem pela API, como, por exemplo, o SDI. Neste caso em específico, o acesso é restrito para leitura, uma vez que o GeoServer deve estar associado à informação geoespacial armazenada.

Com este nível de abstração consegue-se perceber que a API é constituída por um grande conjunto de componentes de gestão de dados. Cada componente tem um papel a desempenhar em determinado momento à medida que os utilizadores interagem com as aplicações.

O componente API Controller é um componente especial que tem como função mapear todos os pedidos feitos à API para os componentes correspondentes e que, por sua vez, irão realizar as operações necessárias.

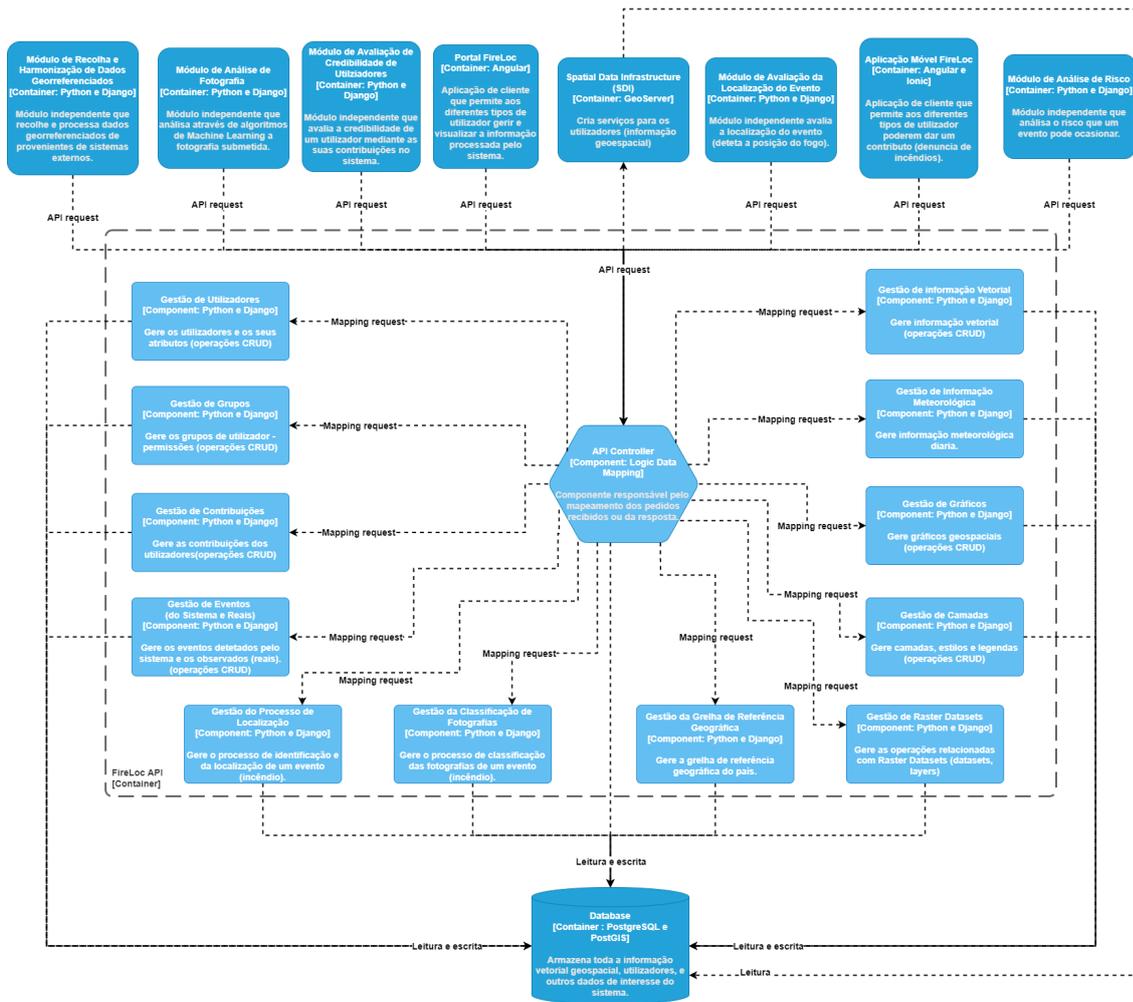


Figura 4.3: Diagrama de Componentes

Apesar de ser possível observar a interação entre containers e componentes, não é especificado qual e que tipo de informação é passada para os componentes. Um bom exemplo é o Módulo de Recolha e Harmonização de Dados Georreferenciados. Neste módulo, em particular, os dados não são processados na API, mas sim no próprio módulo. Consequentemente, implica que a API sirva de ponte para o armazenamento na base de dados e não como meio para o mapeamento para um componente.

4.2 Modelo C&C - Componentes e Conectores

Dando continuidade à descrição da arquitetura e, tendo em conta o exemplo dado na secção anterior sobre a ausência de uma perceção clara do workflow da informação partilhada entre componentes e containers, é apresentada uma visão arquitetural baseada no modelo C&C - Components and Connectors.

Esta vista arquitetural será personalizada com duas outras vistas arquiteturais (3-Tier e Shared Data) de maneira a que o sistema seja dividido em 3 camadas lógicas/físicas do ponto de vista arquitetural com a particularidade de permitir compreender como é que os dados são partilhados pelas camadas e pelos componentes que as constituem através de conectores, sendo estes links de comunicação e/ou protocolos de comunicação.

Apesar desta vista arquitetural ser muito semelhante ao diagrama de container do modelo

C4, consegue identificar aspetos muito úteis do ponto de vista de implementação e que servirá de suporte para a equipa técnica.

4.2.1 3 Tier - Shared Data View

Antes de descrever a vista arquitetural C&C com 3-Tier e Shared Data é conveniente perceber o tipo de notação utilizada para a representação da mesma.

Sendo esta arquitetura baseada em componentes e conectores é prudente destacá-los na arquitetura. Os componentes serão representados com caixas, as cores destacam o tipo de componente que se está a considerar, como se pode observar na figura 4.4. É especificado, também, os links de comunicação que podem ser protocolos de comunicação ou links diretos com a informação que flui sobre o mesmo.



Figura 4.4: Notação 3 Tier - Shared Data View

Client Tier

É importante mencionar que a arquitetura está dividida por camadas lógicas e/ou físicas, como foi revelado no início do Capítulo. Deste modo, podemos destacar a camada Client Tier, que diz respeito às aplicações de interação direta com o utilizador, neste caso, a aplicação móvel e o portal.

Logic Tier

A camada Logic Tier diz respeito a toda a lógica funcional do sistema, pelo que, será constituído por todos os módulos de processamento de informação, assim como os serviços REST e WebSockets, e ainda pelo SDI. Alguns módulos representados na arquitetura do Modelo C4 foram omitidos, nomeadamente, os módulos de gestão de dados, pois fazem parte da API REST. Nesta camada são apenas representados os módulos que eventualmente serão invocados a medida que forem sendo necessários.

Data Tier

Por fim, a camada Data Tier que é constituída pela base de dados do sistema. A camada de dados comunica com a camada lógica através de uma conexão entre a API e a base de dados. A API é o único container responsável pela leitura e escrita de dados.

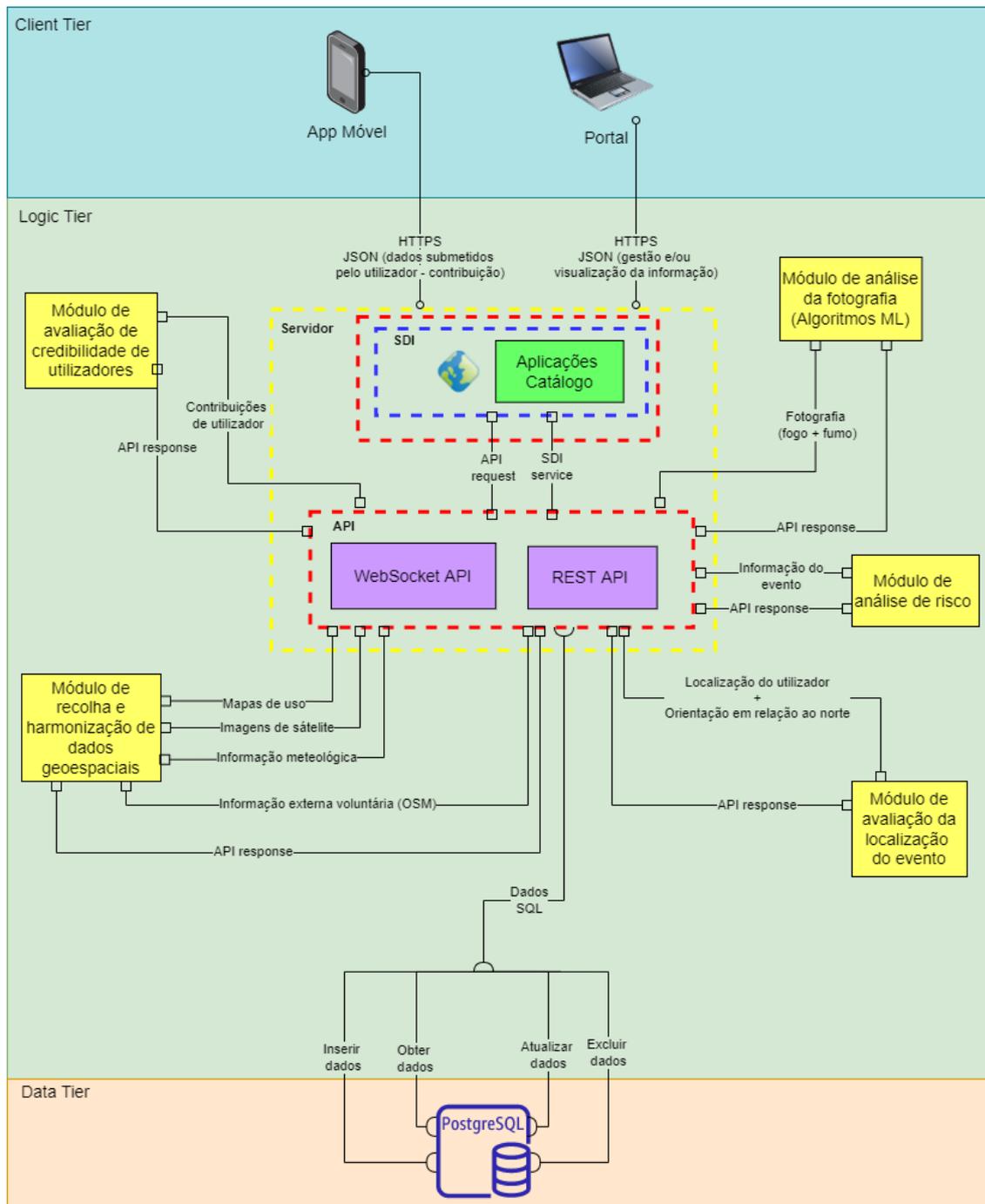


Figura 4.5: Diagrama C&C - 3-Tier com Shared Data

4.3 Comparação entre Modelos

Como foi mencionado no início da secção anterior, foram retirados aspetos úteis do Modelo C&C em relação ao Modelo C4. É importante mencionar que esta vista arquitetural (C&C) está mais direcionada para a equipa de desenvolvimento do que para outro tipo de stakeholder.

Nesta secção, será realizada uma breve comparação entre ambas arquiteturas, nomeadamente, o Modelo C4 e o Modelo C&C (Shared Data View), na qual serão destacadas as suas diferenças e em que contexto cada um dos modelos é útil, seguida de uma breve

justificação.

O Modelo C4 é uma das arquiteturas mais utilizadas para a representação de diferentes níveis de abstração. Cada um dos seus níveis tem um determinado público alvo com o objetivo de desestruturar o sistema em pequenas partes, neste caso representadas em Containers e Componentes. Esta arquitetura é útil para representar a constituição do sistema, isto é, que tipos de utilizadores e sistemas existem, que componentes fazem parte de um determinado container (aplicação), entre outros. Contudo, esta arquitetura não esclarece como é que o sistema funciona, apenas como é constituído.

O Modelo C&C é baseado em componentes e conectores e vem cobrir o problema de especificação de funcionamento do sistema. É evidente a importância que tem em termos de desenvolvimento do sistema, pois especifica a estrutura e o comportamento dos elementos de software, assim como a relação entre elementos. Se adicionarmos ao Modelo a vista Shared Data e a estrutura 3-Tier (Estrutura em camadas), estaríamos a construir uma arquitetura muito completa. A representação em camadas permite separar os componentes do sistema e associá-los a uma camada, seja ela de cliente, lógica ou de dados. Ainda, a vista Shared Data permite compreender a troca e a partilha de dados entre componentes o que facilita ao developer idealizar uma solução para o sistema. Todos os componentes recebem e devolvem dados, e o fluxo de informação é contínuo e deve ser realizado pelos conectores (links).

Desenvolver módulos especificados no Modelo C&C com 3-Tier e Shared Data será muito mais simples por estar detalhado o comportamento e o papel dos módulos no sistema FireLoc, assim como idealizar uma solução para o desenvolvimento das funcionalidades o que, por sua vez, permite reduzir o custo de desenvolvimento e facilita a deteção de erros em termos de dados.

Apesar da utilidade do Modelo C&C, o Modelo C4 consolidou a ideia generalizada que se tem do sistema, permitindo, também, fazer uma separação entre módulos independentes (representados na arquitetura C&C) e os módulos diretos com operações CRUD (representados na arquitetura C4 - Diagrama de Componentes).

4.4 Validação da Arquitetura - Traceability

Para validar a arquitetura do sistema foi construída uma matriz de rastreabilidade. Existem vários tipos e variantes de matrizes Traceability, no entanto, para simplificar a validação da arquitetura, apenas será apresentada a relação entre as funcionalidades do sistema com os UCs apresentados no Capítulo 3. O propósito é identificar a relação entre algumas funcionalidades e UCs. Dada a complexidade e dimensão do projeto serão apresentadas funcionalidades generalizadas e/ou que tenham mais do dois módulos associados.

Use Case	Funcionalidade				
	Autenticar no sistema	Gerir utilizadores no sistema	Identificar incêndios	Validar identificação de incêndios	Processar e analisar dados geospaciais
UC 1	×	×	×	×	×
UC 2	×	×			
UC 3	×	×			
UC 4	×			×	×
UC 5	×		×		×
UC 6	×				×
UC 7	×				×
UC 8	×		×	×	×
UC 9	×				×
UC 10	×				×
UC 11	×			×	×

Tabela 4.1: Traceability - Relação entre Funcionalidades e UCs

Como se pode observar na tabela 4.1, é possível constatar uma relação entre algumas funcionalidades do sistema com os seus módulos, isto é, os UCs. É natural que todos os UCs tenham uma relação com a funcionalidade de Autenticar no sistema, uma vez que para ter acesso aos módulos é necessário validar as credenciais do utilizador.

Em relação à funcionalidade de Gerir utilizadores ao sistema é notável que tem três UCs associados. Isto deve-se pelo facto desta funcionalidade implicar a associação obrigatória de um utilizador a um grupo (UC 3).

A identificação de incêndios está associada à Gestão de Eventos (UC 5), à Gestão de Resultados do Processo de Localização de Eventos (UC 8) e à Gestão dos Resultados do Processo de Classificação de Fotografias (UC 11), uma vez que, a identificação de incêndios propriamente dita implica a validação da ocorrência através de algoritmos o que, por sua vez, gera um resultado de validação. A identificação de incêndios não é imediata e deve ser realizado um processo para o efeito, o qual pode ser seguido e gerido.

A validação da identificação de incêndios está relacionada com a Gestão de Contribuições (UC 4), visto que as contribuições terão de ser, obrigatoriamente, validadas para que possam ser consideradas pelo sistema FireLoc. Neste processo de validação, as contribuições serão avaliadas (UC 8) e as fotografias da mesma classificadas (UC 11) por algoritmos de Machine Learning. Estes algoritmos não são alvo de estudo nesta dissertação pois, uma equipa do FireLoc está a dedicar-se, exclusivamente, nestes algoritmos.

Por fim, processar e analisar dados geospaciais está associada a quase todos os módulos da API, que tem sido o principal foco de trabalho.

Na tabela 4.1 houve um foco nas principais funcionalidades do sistema e que podem ser agrupadas por categorias funcionais. As categorias da tabela (funcionalidades) são genéricas o suficiente para abranger o maior número de UCs. Visto que todos os UCs conseguiram estar relacionados com as funcionalidades, pode-se concluir que a arquitetura está bem estruturada, o que por sua vez nos permite afirmar que está validada.

Capítulo 5

Implementação

Nos Capítulos anteriores, foram apresentadas as funcionalidades do sistema e a sua arquitetura de software, assim como a relação entre ambos. Partindo desta informação e dos objetivos definidos nesta dissertação, será descrito e detalhado todo o processo de implementação dos módulos.

O Capítulo será dividido por categoria tecnológica e componente desenvolvido, abordando-se um tópico diferente de desenvolvimento em cada secção. Cabe destacar que todas as categorias tecnológicas estão diretamente relacionadas e houve a necessidade de integrar alguns dos módulos e componentes desenvolvidos como iremos descrever mais em frente.

5.1 Desenvolvimento da API

Para o desenvolvimento da API foi escolhido o framework Django, sendo baseado em aplicações que podem ser consideradas packages reutilizáveis. As aplicações podem ser desenvolvidas de raiz ou instaladas de fontes externas. Neste momento o sistema FireLoc conta com uma série de aplicações desenvolvidas de raiz e aplicações externas que serviram como ferramentas de suporte para o desenvolvimento da API.

5.1.1 Aplicações Externas

Entre as aplicações externas podemos destacar aquelas utilizadas, efetivamente, para o desenvolvimento da API, sendo uma dessas aplicações o DB Backup. Esta aplicação é um package que permite utilizar comandos para automatizar o processo de backup e restaurar a base de dados do sistema FireLoc, assim como dos ficheiros utilizados no sistema, como, por exemplo os ficheiros Raster, entre outros. A aplicação Django Rest Framework é outra aplicação externa e a mais importante para o desenvolvimento da API, uma vez que serviu de base para o desenvolvimento dos endpoints. É uma aplicação desenhada especificamente para o desenvolvimento web de APIs e que se integra facilmente com o OAuth2. O OAuth2, apesar de não ser uma aplicação, é um protocolo de autorização utilizado para o desenvolvimento de endpoints e que ajudou na proteção dos mesmos com o sistema de autenticação e autorização. Por último, Cors Headers é uma aplicação que nos permite aceder à API a partir de outros domínios, sendo essencial para o desenvolvimento de endpoints, uma vez que é necessário aceder a API e a outros domínios locais ou públicos para realizar os testes de desenvolvimento.

5.1.2 Aplicações Internas

Por outro lado, as aplicações internas são aquelas em que o aluno se focou e desenvolveu componentes. Por essa razão e, pelo contexto do capítulo, são apenas descritas as aplicações em que o estagiário realizou as suas atividades de implementação e são omitidas aquelas nas quais não houve qualquer tipo de interação. Entre as aplicações implementadas, podemos destacar as aplicações:

- Authapi — relacionada com o desenvolvimento de três módulos: Gestão de Utilizadores, Gestão de Grupos e o Sistema de Autenticação e Autorização do sistema;
- Contrib — relacionada com o desenvolvimento do módulo da Gestão de Contribuições do sistema;
- Events — relacionada com o desenvolvimento do módulo da Gestão de Eventos (Reais e Áreas Ardidas);
- Firedetect — relacionada com dois módulos: Gestão dos Resultados do Processo de Localização de Eventos e a Gestão dos Resultados do Processo de Classificação de Fotografias que dizem respeito aos Eventos identificados pelo Sistema;
- GeoRef — relacionada com o módulo de Gestão de Unidades Territoriais e o módulo de Gestão da Grelha de Referência Geográfica;
- GeoRst — relacionada com o módulo de Gestão de Dados Raster;
- GeoVis — relacionada com três módulos: Gestão de Camadas, Gestão de Legendas e a Gestão de Gráficos;
- Meteo — relacionada com o módulo de Gestão de Informação Meteorológica;
- SDI — relacionada com o módulo de Gestão do SDI, isto é, de todas operações que se realizam com o GeoServer.

É de notar que estas aplicações estão relacionadas com todos os módulos apresentados nas Funcionalidades e Arquitetura do Sistema. Nas tabelas que se seguem, estão identificadas as aplicações mencionadas.

Aplicações Externas do Django
DB Backup
Django Rest Framework
OAuth2
Cors Headers

Tabela 5.1: Aplicações externas - Django

Aplicações Internas do FireLoc
Authapi
Contrib
Events
Firedetect
GeoRef
GeoRst
GeoVis
Meteo
SDI

Tabela 5.2: Aplicações internas - Sistema FireLoc

As aplicações internas identificadas na tabela 5.2 serão descritas com maior detalhe, assim como, os elementos de desenvolvimento envolvidos em cada uma delas nas secções seguintes.

5.1.3 Boas Práticas utilizadas

Para o desenvolvimento da API do sistema FireLoc foi seguido um conjunto de regras que induzem às boas práticas no desenvolvimento de qualquer API. As boas práticas, a seguir representadas, já se encontravam definidas pela equipa de desenvolvimento do sistema FireLoc, de modo que, o desenvolvimento dos módulos atuais e futuros, sigam a mesma estrutura e seja mais simples a revisão e manutenção de código. Entre as boas práticas podemos destacar as seguintes:

1. Usar o formato JavaScript Object Notation (JSON). Este formato é o mais utilizado para enviar e receber dados através de APIs pela sua legibilidade, facilidade de manipulação e por ser um formato de dados lightweight.
2. Uso de nomes nos endpoints. A API do sistema FireLoc não utiliza verbos para destacar os seus endpoints. Na verdade, utiliza nomes com relevância no contexto da aplicação em questão.

```
# Metodo HTTP: GET
http://www.example.com/getUsers # Má prática
http://www.example.com/users # Boa prática

# Metodo HTTP: POST
http://www.example.com/createUser # Má prática
http://www.example.com/users # Boa prática
```

Bloco 5.1: Uso de nomes em endpoints - Exemplo

Esta prática é correta uma vez que os verbos nos endpoints podem ser omitidos por estar implícito o verbo propriamente dito com os métodos HTTP. A vantagem desta prática é evitar criar um URL para cada endpoint, isto é, definir apenas um URL para um conjunto de endpoints na qual só é alterado o método HTTP. No bloco de código 5.1 podemos observar que um URL (endpoint) pode ser usado para vários métodos evitando, assim, o uso de verbos nos URLs.

3. Uso de Códigos na representação das respostas dos pedidos à API. O sistema FireLoc baseia-se em “Status Codes” para a representação das suas respostas. Na tabela 5.3, pode-se observar um exemplo dos códigos utilizados para representar o sucesso das respostas nos pedidos à API.

Resposta HTTP	Código	Descrição
200	S20	Data successfully returned
201	S21	New created
201	S22	Was updated
200	S23	Was deleted
200	S24	All were deleted
201	A21	Acess tokens were generated
201	A22	Confirmation token was generated
201	G21	Was created
201	G22	Was updated
201	G23	Was deleted

Tabela 5.3: Códigos de Sucesso

4. Uso de Secure Socket Layer (SSL) para a proteção da API. O uso de SSL permite ao sistema FireLoc garantir que a informação entre a API e os browser será encriptada. É claro que, para esta fase de desenvolvimento dos módulos, o ambiente operacional não inclui esta camada de segurança. O uso de SSL foi tido em conta, apenas, no modo de produção.
5. Documentar a API. A documentação da API do sistema FireLoc é, naturalmente, privada para a equipa de desenvolvimento. Contudo, a documentação existe e tem sido de grande suporte. A documentação é muito importante em qualquer projeto, de maneira a explicar como utilizar o sistema e auxiliar qualquer membro da equipa na compreensão do mesmo.

No projeto FireLoc existem diferentes tipos de documentação. A documentação relacionada com toda a parte de âmbito e negócio, a documentação de desenvolvimento (GitHub) e a documentação da API. A documentação da API foi realizada recorrendo ao Jupyter Lab e à plataforma OpenAPI/Swagger. Estas duas aplicações serão descritas com mais detalhe mais em frente.

6. Uso de Serializers. Os Serializers permitem-nos, não só realizar as operações CRUD à base de dados, como, também, validar a estrutura dos dados recebidos, nomeadamente, na criação de objetos e na atualização dos mesmos. Este aspeto será mais claro com a descrição do processo de desenvolvimento em cada uma das aplicações.
7. Uso de ORM do Django. O uso do ORM facilita muito a interação com a Base de Dados, nomeadamente, nas operações CRUD e na manipulação de tabelas, campos e restrições, como a sua criação, atualização, remoção, entre outros.
8. Uso de Docker para independizar aplicações. O uso do Docker como tecnologia de virtualização, permite isolar aplicações e independizar o seu funcionamento e, desta forma, tornar o sistema modular, baseado na integração de componentes fragmentados. As atividades com Docker foram um dos objetivos idealizados para o estágio, sendo descritas mais em frente.

Nesta secção, é apresentado o processo de desenvolvimento realizado para cada uma das aplicações internas do sistema FireLoc seguindo as boas práticas acima descritas.

5.1.4 Authapi

Como foi revelado anteriormente, a aplicação Authapi é constituída por três Módulos: a Gestão de Utilizadores, a Gestão de Grupos e o Sistema de Autenticação e Autorização do sistema. A descrição do processo de desenvolvimento será semelhante em todos os módulos no qual se discute a estrutura dos objetos manipulados e técnicas usadas para o desenvolvimento. Todos os endpoints da aplicação Authapi, assim como, das restantes aplicações internas podem ser encontrados no Apêndice C. Em cada módulo serão descritos alguns detalhes de desenvolvimento.

Sistema de Autenticação e Autorização

O módulo relacionado com o Sistema de Autenticação e Autorização é constituído por cinco endpoints, e é baseado na gestão de Tokens. Os Tokens, controlados pela aplicação externa OAuth2, permitem que seja realizado o acesso às funcionalidades da aplicação caso estes sejam válidos para o nosso sistema funcionando como um middleware, de contrário,

o acesso será negado. No Capítulo 6 serão apresentados os testes realizados em termos de segurança da API onde se observa os efeitos de segurança deste módulo.

O desenvolvimento dos endpoints deste módulo tiveram algum nível de complexidade, nomeadamente, na integração da package OAuth2 e no envio de e-mails. Para a integração do protocolo OAuth2, foi necessário registar o sistema FireLoc, o qual tem Tokens confidenciais da aplicação, o Client ID e o Secret ID. Com estes Tokens, a aplicação deve gerar Tokens de Acesso aos seus utilizadores quando estes os solicitarem. Contudo, se o utilizador não estiver registado no sistema FireLoc, o Token de Acesso não é enviado. Para utilizadores registados, o Token de acesso permite que tenham acesso à grande maioria das funcionalidades, dependendo das suas permissões (grupo).

Na prática, para que o sistema interprete que o Token de Acesso é válido e permita aos seus utilizadores terem acesso às funcionalidades, os pedidos devem enviar o Token com um esquema definido. O esquema utilizado no sistema FireLoc é o Bearer Authentication, que é reconhecido pelos pedidos HTTP como uma referência à autorização do portador.

A validação do Token pelo protocolo OAuth2 é feito pela própria package em si. A API do FireLoc apenas se limita a reencaminhar o pedido de autenticação e/ou autorização aos servidores do OAuth2. Cabe destacar que o Token tem um período de validade, isto é, o token pode expirar passado um tempo definido.

Os Tokens de acesso acabam por ser uma camada de segurança para a API, complementar às classes de autorização definidas em todos os endpoints. As classes de autorização são grupos de utilizadores autorizados, isto significa que, se um dado endpoint não tiver uma classe associada, esse grupo de utilizador não terá acesso ao endpoint.

Este módulo tem alguns componentes relacionados ao envio de e-mails, nomeadamente, os endpoints para confirmação do registo de um utilizador e para reencaminhar um utilizador à página dedicada para recuperação de password. Para o sistema FireLoc enviar e-mails foi utilizada uma biblioteca para o efeito, o `smtplib`, baseada no protocolo Simple Mail Transfer Protocol (SMTP). A biblioteca é simples no contexto de implementação, contudo, é necessário realizar alguns passos de configuração adaptados ao servidor do FireLoc, nomeadamente, o porto 587 dedicado para conexão e transporte seguro, o servidor SMTP `smtplib@gmail.com` e o e-mail oficial do grupo FireLoc.

Gestão de Utilizadores

O módulo de Gestão de utilizadores, constituído por sete endpoints, está diretamente relacionado com outros módulos, como a Gestão de Grupos e a Gestão de Atributos do Utilizador. O Módulo de Gestão de Atributos do Utilizador é um módulo complementar aos utilizadores que apenas se limita a registar e controlar os atributos extras dos utilizadores, como a idade, profissão, entre outros. Neste e em todos os outros módulos são realizadas uma série de verificações de modo a que o sistema tenha sempre resposta a qualquer situação, como certificar se os dados são válidos, existência de múltiplas instâncias, tentativa de duplicar dados, entre outros. Esta informação estará implícita nos restantes módulos que se apresentarão uma vez que é semelhante em todos os endpoints e é um processo natural no desenvolvimento de qualquer API.

Para o desenvolvimento dos endpoints, neste módulo e nos restantes, foram usados serializers como foi revelado na secção 5.1.3. Os serializers são ótimos para a validação da estrutura dos dados recebidos. Para fazer uma inserção (POST), por exemplo, o serializer verifica se a estrutura dos dados é válida a partir dos modelos definidos. Caso a estrutura

não seja respeitada, a operação não é realizada. Isto é muito importante para garantir que o fluxo dos dados siga uma ordem e estrutura bem definida. Esta prática implicou desenvolver um serializer para cada modelo da base de dados.

Os serializers podem, no entanto, ter outros serializers associados. Dito isto, é natural, por exemplo, que o serializer do modelo de utilizadores tenha associado um serializer do modelo dos atributos de utilizador.

Um exemplo na qual são manipulados vários serializers é na inserção de utilizadores. Na inserção de um utilizador tem de se ter em conta o grupo ao qual este utilizador ficará associado e os atributos extras que podem ser enviados no pedido. Dependendo do endpoint, o atributo referente ao grupo poderá ser obrigatório ou não. Para a criação de utilizadores regulares (UC 1.2) não é obrigatório uma vez que por ser regular, o seu grupo já está pre-definido.

A gestão de atributos é constituída por dois modelos, um para controlar os atributos e outro para controlar os seus valores e a associação com os utilizadores. Por essa razão, a inserção de atributos de um utilizador é um processo mais complexo uma vez que é necessário manipular mais do que um modelo (Modelo de utilizadores, grupos, atributos e valores de atributos).

Gestão de Grupos

O módulo de Gestão de grupos é constituído por seis endpoints. As classes e objetos manipulados para a gestão de grupos são provenientes dos modelos nativos do Django, isto significa que os modelos não foram criados pela equipa do FireLoc. O mesmo acontece com o Modelo dos Utilizadores e as permissões associadas a cada grupo.

O desenvolvimento dos endpoints para a Gestão de Grupos não tiveram qualquer nível de complexidade uma vez que se limitam à gestão da informação. Cada grupo tem um conjunto de permissões associados que, na sua versão mais primitiva, são uma associação a uma dada operação de leitura e escrita. Para que fique mais clara a associação entre operações e permissões, cada módulo terá, em média, quatro permissões diretamente relacionadas com os quatro métodos HTTP:

- Pode ver;
- Pode adicionar;
- Pode alterar;
- Pode apagar.

Estas permissões são registadas, automaticamente, pelo Django para cada modelo criado. Cabe ao sistema FireLoc atribuir, no entanto, estas permissões a cada grupo.

5.1.5 Contrib

Dando continuidade à descrição do desenvolvimento das aplicações internas, segue-se a aplicação Contrib. Esta aplicação é constituída por apenas um módulo: a Gestão de Contribuições. Este módulo envolve a manipulação de tipos de dados complexos, nomeadamente, dados geospaciais e fotografias.

Gestão de Contribuições

O módulo de Gestão de Contribuições, constituído por sete endpoints, está relacionado com o processamento de informação geospacial e armazenamento de fotografias. Estes dados são enviados pelos voluntários, o que torna evidente a importância deste módulo para sistema, uma vez que, gere a nossa principal fonte de dados.

Para o desenvolvimento dos endpoints foram usadas bibliotecas GIS, podendo-se destacar o Shapely e o GDAL, para o cálculo e manipulação de projeções e reestruturar as geometrias enviadas pelos utilizadores nas suas contribuições.

O sistema FireLoc tem uma wrapper destinado para o desenvolvimento de métodos e/ou algoritmos para a manipulação de dados geospaciais, o Firegis. O Firegis acaba por ser uma package adaptada resultando na integração de outras bibliotecas de suporte para o processamento de informação geospacial, como o GDAL, GRASS e o Shapely, entre outros. Dito isto, cabe realçar que o Firegis é a principal package usada para todos os processos que envolvam dados georreferenciados.

Como já foi revelado, o principal foco de trabalho é o processamento de informação geospacial e, por esse motivo, será descrito alguns dos passos mais comuns realizados no desenvolvimento de componentes que implicaram trabalhar com este tipo de dados.

Um dos passos mais importante é verificar a projeção dos dados geospaciais, pois a informação pode não estar com as condições pretendidas em termos de representação. Uma projeção é o conceito utilizado para nos referirmos aos tipos de representação de uma determinada geometria geospacial. Como é sabido, o planeta Terra tem uma superfície curva, no entanto, os mapas são representados em superfícies planas. Por esta razão é necessário realizar uma transformação através de cálculos matemáticos, de modo que seja possível converter o sistema de coordenadas usado na superfície curva da Terra para uma superfície plana. As projeções, neste caso, serão realizadas para o EPSG 3763 que corresponde ao CRS português.

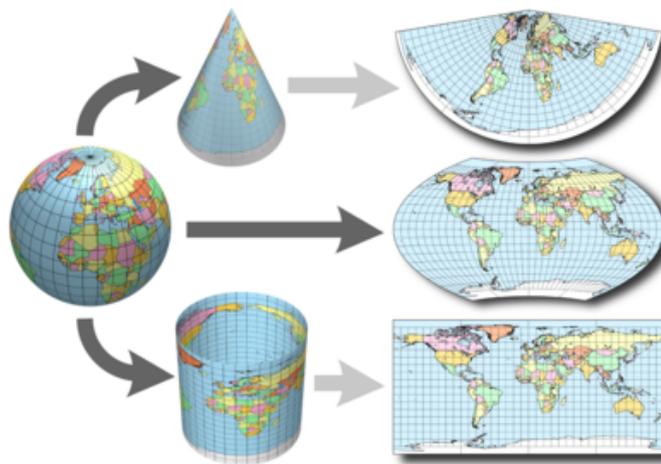


Figura 5.1: Diferentes tipos de projeções [67]

Na figura 5.1 podemos observar os diferentes tipos de projeções. Como não existe uma transposição de superfície curva para uma superfície plana sem alguma distorção, existem várias projeções de mapas que fornecem propriedades diferentes, alguns preservam a forma, enquanto outros preservam a distância, entre outros [7].

Outro passo importante é a transformação adequada da geometria geospacial. Na grande

maioria dos casos, o que é enviado e recebido não são as geometrias propriamente ditas, mas sim uma representação simplificada das mesmas. Para ficar mais claro, será introduzido o conceito de Well-known text (WKT) que incide na representação de geometrias vetoriais através de texto [87]. É reconhecido pela OGC, o que faz dela uma representação bastante usada, contudo, é necessário verificar se o WKT é válido e corresponde a uma geometria reconhecida para que, posteriormente, seja aplicada uma transformação da representação textual para objeto geométrico.

Tipo	Exemplos	
Apontar		POINT (30 10)
LineString		LINESTRING (30 10, 10 30, 40 40)
Polígono		POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))
		POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))

Figura 5.2: Representação em texto de geometrias vetoriais 2D - WKT [87]

Como se pode observar na figura 5.2, estão representados alguns exemplos de WKT em relação à algumas geometrias. O GDAL foi usado para aplicar as transformações. Estes passos/processos são realizados em todos os endpoints que envolvam geometrias e, portanto, estarão implícitos nos restantes endpoints que venham a ser descritos.

Em relação ao processamento das fotografias enviadas na contribuição, as fotografias são guardadas no servidor do FireLoc, assim como a referência às mesmas na base de dados através do seu caminho relativo, de forma a poder aceder às fotografias pelos serviços REST. Para guardar as fotografias no servidor foi necessário realizar uma série de operações, uma vez que a fotografia deve ser transformada em bytes. Para tal foi usado o Base64 Encode para ambas operações, transformação de imagem para bytes e vice-versa.

5.1.6 Events

A aplicação Events está relacionada com a Gestão de Eventos (incêndios). Os eventos como já foi revelado no Capítulo 3, são de dois tipos: Eventos Reais e Eventos do Sistema. Por esse motivo, o módulo de Gestão de Eventos está na verdade dividido em 3 componentes, a Gestão de Eventos Reais, a Gestão de Áreas Ardidias, e a Gestão de Anos, sendo este último apenas um módulo complementar.

É importante referir que os eventos identificados pelo sistema não são abordados na aplicação Events, mas sim na aplicação Firedetect, tratada na seguinte secção (Secção 5.1.7).

Gestão de Eventos Reais

O módulo de Gestão de Eventos Reais, constituído por seis endpoints, diz respeito a eventos observados à priori, isto é, eventos que não foram processados ou previstos, mas que aconteceram em algum momento na vida real. Estes eventos estão associados a um ano específico que corresponde ao ano da ocorrência - Gestão de Anos.

Toda esta informação é extraída de fontes externas, sendo neste caso em específico dados

retirados do Catálogo Geográfico do ICNF, que contém informação geográfica disponível para download em diferentes formatos. Para o registo de eventos reais estão a ser usado os datasets disponibilizados em [17].

Estes datasets têm vários ficheiros que, no seu conjunto, representam dados geospaciais. Contudo, estes ficheiros, por si sós, não podem ser interpretados nem lidos de uma maneira direta e, por tanto, foram desenvolvidos scripts auxiliares para extrair a informação dos datasets.

Nos scripts desenvolvidos são usados, novamente, métodos do Firegis. Como foi dito, estes ficheiros não podem ser lidos de uma maneira direta e, por esse motivo, foi necessário transformar o ficheiro num formato legível e tornar possível a manipulação do seu conteúdo. É um passo imprescindível conhecer o EPSG dos dados em questão. Neste caso, como os dados vêm do ICNF terão, em princípio, o EPSG adequado. Contudo, é sempre boa prática verificar e realizar as projeções e transformações caso não se verifique.

Gestão de Áreas Ardidas

O módulo para a gestão de áreas ardidas, constituído por seis endpoints, é, como o próprio nome indica, um registo das áreas ardidas no território português ao longo dos anos - Gestão de Anos. Este módulo não deve ser confundido com a Gestão de Eventos Reais, pois, apesar de ambos fazerem referência a eventos observados, as áreas ardidas focam-se na identificação dos territórios afetados e no número de vezes em que determinada área ardeu. Estes dados são muito úteis em termos de prevenção e gestão de incêndios, uma vez que permite identificar as áreas em que ocorram incêndios mais frequentemente e construir um plano de prevenção e de atuação para essas áreas.

No contexto do FireLoc, não houve necessidade de registar a contagem em que uma determinada área foi afetada, por outro lado, houve a necessidade de registar o ano em que a mesma ardeu. Esta informação permite-nos inferir quais são as áreas mais propícias a sofrer incêndios e outras que não (Gestão de Risco de Incêndio), visto que, como foi explicado no Capítulo 2, o tecido florestal necessita de tempo para se regenerar e não faz sentido considerar áreas que tenham sido afetadas recentemente uma vez que não haverá vegetação para que um incêndio ocorra. Tal como a Gestão de Eventos Reais, estes dados são obtidos a partir do catálogo do ICNF.

5.1.7 Firedetect

A aplicação Firedetect está relacionada com a Gestão de Resultados do Processo de Localização de Eventos, constituída por vinte e quatro endpoints, e a Gestão de Resultados do Processo da Classificação de Fotografias, constituída por dezoito endpoints no total. Esta aplicação diz respeito aos eventos identificados pelo sistema FireLoc e não deve ser confundida com a aplicação Events. Estes módulos são bastantes complexos no que diz respeito à relação entre os seus modelos por haver bastantes dependências entre eles.

Gestão de Resultados do Processo de Localização de Eventos

O módulo de Gestão de Resultados do Processo de Localização de Eventos tem uma grande quantidade de endpoints, pois depende de muitos componentes. Os atributos, abordagens e procedimentos estão diretamente associados aos resultados do processo. Deste modo, para que um resultado seja registado, por exemplo, terá de haver registos dos outros modelos.

Todos os endpoints são bastante simples, com exceção aos endpoints para a gestão dos resultados pelas razões acima apresentadas. Estes endpoints limitam-se a controlar o fluxo de trabalho realizado para a deteção de incêndios com scripts, como foi descrito no Capítulo 3, na secção 3.1.8.

Gestão de Resultados do Processo da Classificação de Fotografias

Em relação ao módulo de Gestão de Resultados do Processo da Classificação de Fotografias, é importante destacar que tem uma organização de endpoints semelhantes à Gestão de Resultados do Processo de Localização de Eventos. O desenvolvimento destes endpoints não tiveram qualquer nível de complexidade. As técnicas utilizadas já foram descritas nos módulos anteriores.

Apesar do desenvolvimento destes endpoints não terem apresentado dificuldades, os conceitos e componentes dos módulos da aplicação Firedetect não eram evidentes. O grande desafio neste módulo foi compreender como é que os diferentes componentes se relacionavam entre si e o papel que desempenham em todo o processo.

5.1.8 GeoRef

A aplicação GeoRef não foi desenvolvida pelo estagiário de raiz, contudo, tem sido usada em outros módulos como iremos ver mais em frente e, por essa razão, é descrita nesta secção. Esta aplicação está relacionada com o módulo Gestão de Unidades Territoriais e o módulo de Gestão da Grelha de Referência Geográfica.

Gestão de Unidades Territoriais

As Unidades Territoriais no sistema FireLoc são a base para a aplicação do modelo 3 da Nomenclatura das Unidades Territoriais para Fins Estatísticos (NUT). Esta nomenclatura representa um território em regiões e, dependendo do modelo em questão, a divisão territorial será mais precisa [64].

A Carta Administrativa Oficial de Portugal (CAOP) regista estas divisões e representa-as através de modelos, como se pode observar na figura 5.3. O modelo III está a ser usado no sistema e permite que a divisão do território seja mais precisa o que, por sua vez, nos ajuda a fazer uma melhor gestão da informação. Contudo, este modelo divide o território em 25 NUTs (unidades territoriais), o que ainda implica que haja um elevado nível de dificuldade no momento de identificar lugares muito específicos. Por esse motivo, o modelo 3 foi, ainda, dividido em concelhos, e os concelhos, por sua vez, em freguesias. Desta maneira, o sistema FireLoc consegue gerir com muita mais precisão os dados geospaciais. Este tipo de informação é útil, pois permite identificar as zonas do país que estão a ser afetadas por incêndios florestais e localizar as áreas de atuação. Cabe destacar que estes dados (NUTs, concelhos e freguesias) são provenientes de fontes externas confiáveis, como o Portal de dados abertos da Administração Pública fornecidos pelo Governo de Portugal em [19].

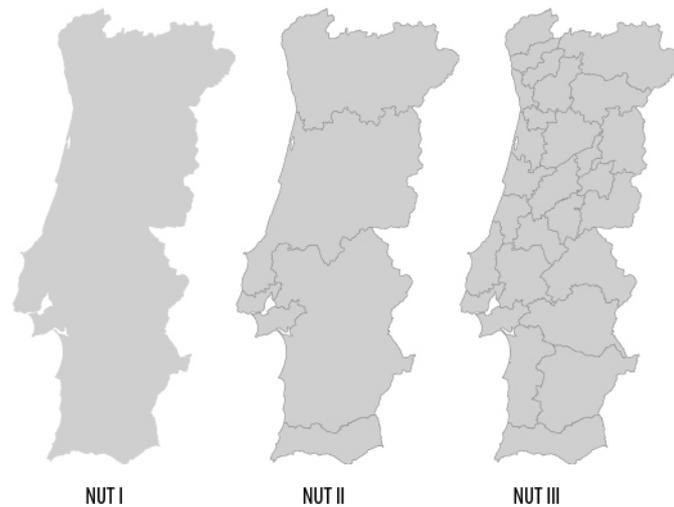


Figura 5.3: Modelos NUTs - Imagem adaptada de [74]

Gestão da Grelha de Referência Geográfica

Por outro lado, para representar determinada informação, o sistema FireLoc utiliza uma grelha de referência aplicada em Portugal, isto é, o território português está dividido em linhas e colunas, formando um conjunto de células.

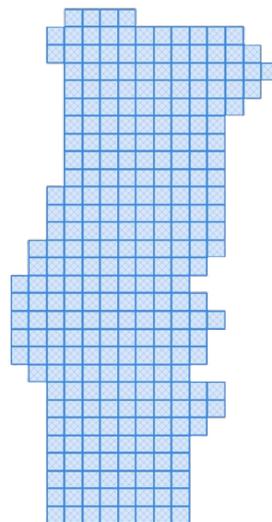


Figura 5.4: Grelha de Referência Geográfica para Portugal Continental

As células são precisamente a referência de localização e estão diretamente associadas a uma geometria geospacial. Todos os dados associados ao módulo estão guardados na base de dados e, através da extensão do PostGIS, serão realizadas queries que tenham em conta a grelha de referência.

5.1.9 GeoRst

Em relação à aplicação GeoRst, é uma aplicação bastante importante, pois é a única que gere e processa dados raster. A aplicação GeoRst é constituída por quatro módulos: o módulo de Gestão de Tipos Raster, Gestão de Datasets Raster, Gestão de Camadas Raster e um endpoint dedicado para adicionar ficheiros Raster.

Gestão de Tipos Raster

O módulo de Gestão de Tipos Raster é constituído por apenas três endpoints. O seu desenvolvimento foi bastante simples, sem apresentar qualquer tipo de dificuldade. É um módulo complementar para a Gestão de datasets raster, no qual podemos distinguir o tipo de dataset que se está a considerar. No Capítulo 2 vimos, por exemplo, os Modelos Digitais, porém, existem outros como Coberturas do solo, entre outros tipos.

Gestão de Datasets Raster

A Gestão de datasets raster está associada ao módulo de Gestão de tipos Raster, uma vez que, cada dataset tem um determinado tipo. Os datasets são, na prática, um conjunto de camadas raster. Para gerir os datasets e conseguirmos controlar os seus dados (camadas raster), foram desenvolvidos seis endpoints tendo sempre em consideração a associação entre modelos.

Gestão de Camadas Raster e Ficheiros Raster

A Gestão de Camadas Raster não envolve, ainda, processamento dos dados e, por isso, não devem ser confundidos os conceitos. Até agora, têm sido, apenas, apresentados módulos para gerir alguma informação associada (tipos e datasets) e que estão relacionadas às camadas. As camadas raster são, tal como nos módulos anteriores, apenas uma referência, servindo como modelo complementar para a organização dos dados geospaciais raster. Contudo, a Gestão de Ficheiros Raster é o módulo responsável pelo processamento dos dados geospaciais raster.

Para clarificar como é que se pode associar um ficheiro raster a uma determinada camada e dataset é necessário introduzir alguns conceitos aplicados neste módulo.

É importante discutir que ficheiros irão ser adicionados ao sistema e como os iremos usar. Como foi explicado no Capítulo 2, mais especificamente no tópico de Tipos de Dados Raster, o COS será uma das principais fontes de dados raster do sistema FireLoc. Recordando o que foi mencionado, os dados raster provenientes do COS permitem identificar que lugares, em Portugal, dizem respeito a zonas urbanas, o tecido florestal, as linhas de água, entre outros. Esta informação é, muito importante no contexto do sistema FireLoc para a identificação, gestão e prevenção de incêndios florestais. Esta informação não pode ser acedida automaticamente e adicionada ao sistema FireLoc e, por essa razão, foram criados os endpoints anteriores de maneira a podermos ter esta informação no nosso sistema.

Para o armazenamento dos dados raster introduziu-se um novo formato de ficheiros, o GeoPackage (GPKG). O GPKG é um formato suportado pela OGC e descreve um conjunto de normas através de uma estrutura baseada em armazenamento de dados SQLite para guardar datasets raster, características de dados geospaciais (features), atributos peculiares,

entre outros. Isto significa que o GPKG consegue ter no seu conteúdo informação relativa a um conjunto de dados geospaciais vetoriais ou raster [31]. Foi desenhado para funcionar como um único ficheiro e ser o mais lightweight possível para o desenvolvimento [81] e, na prática, funciona como uma tabela de uma base de dados. A razão de se ter desenhado assim é pelo facto de se especializar na adoção para o desenvolvimento de aplicações móveis e facilitar a organização e acesso de dados raster/vetoriais [31].

Introduzir um novo formato de ficheiros pode parecer escusado, contudo, o objetivo é evitar gerir uma grande quantidade de ficheiros. Na descrição do módulo de Gestão de Eventos Reais foi mencionado que os datasets de dados têm vários ficheiros que atuam em conjunto e, portanto, não podem ser desvinculados se mantivermos os formatos originais. Com o uso do GPKG irá trabalhar-se apenas com 1 ficheiro o que facilita muito o controlo de informação e acesso.

Para armazenar a informação em GPKG foi desenvolvido um script com métodos do Firegis, com recurso, principalmente, do GDAL.

Apesar do GDAL ter estas funcionalidades prontas para uso, é necessário primeiro realizar uma série de verificações aos dados raster. Nos módulos em que se processa informação vetorial, têm sido feitas verificações nas projeções. Com os dados raster deverá ser feito o mesmo processo de modo a garantir as projeções corretas.

Neste módulo, são utilizados os valores dos vértices de cada uma das células dos raster para garantir que são válidos na Grelha de Referência do sistema FireLoc, descrita na secção 5.1.8. Por outras palavras, devemos obter as dimensões das células raster e verificar se essa célula corresponde a uma célula da Grelha de Referência. Caso a célula raster não tenha correspondência com alguma célula da grelha, o ficheiro não é adicionado ao sistema.

Para realizar esta verificação foi construída uma query Structured Query Language (SQL) complexa à base de dados. Nas primeiras tentativas verificou-se que as coordenadas dos vértices da célula do raster nunca coincidiam com nenhuma célula da grelha. O motivo pelo qual isto acontecia era pelo facto das coordenadas do raster terem demasiada precisão (cerca de 15 casas decimais) e as coordenadas das células da grelha terem menos sensibilidade (cerca de 7 casas decimais). A sensibilidade do PostGIS não pode ser alterada e, por tanto, houve a necessidade de contornar este problema. Para resolver esta questão, implementou-se uma estratégia na qual se compara se as coordenadas dos vértices da célula do raster e os vértices da célula da grelha são inferiores a 0.000001 (≈ 0).

```
q = ("SELECT * FROM ("
  "SELECT tgrid.gid, tgrid.cellid, "
  "ST_Distance(ST_SetSRID(ST_MakePoint("
    f"ST_XMin({geomcol}), ST_YMax({geomcol})), {int(tblsrs)}), "
    f"ST_GeomFromText('POINT ({str(left)} {str(top)} 0)', {int(epsg)})"
  ") AS dist_topleft, "
  "ST_Distance(ST_SetSRID(ST_MakePoint("
    f"ST_XMax({geomcol}), ST_YMax({geomcol})), {int(tblsrs)}), "
    f"ST_GeomFromText('POINT ({str(right)} {str(top)} 0)', {int(epsg)})"
  ") AS dist_topright, "
  "ST_Distance(ST_SetSRID(ST_MakePoint("
    f"ST_XMax({geomcol}), ST_YMin({geomcol})), {int(tblsrs)}), "
    f"ST_GeomFromText('POINT ({str(right)} {str(bottom)} 0)', {int(epsg)})"
  ") AS dist_lowright, "
  "ST_Distance(ST_SetSRID(ST_MakePoint("
    f"ST_XMin({geomcol}), ST_YMin({geomcol})), {int(tblsrs)}), "
    f"ST_GeomFromText('POINT ({str(left)} {str(bottom)} 0)', {int(epsg)})"
  ")")
```

```
) AS dist_lowleft "  
  f"FROM {tbl} AS tgrid"  
) AS foo "  
"WHERE dist_topleft < 0.000001 AND dist_topright < 0.000001 "  
"AND dist_lowright < 0.000001 AND dist_lowleft < 0.000001"
```

Bloco 5.2: PostGIS Query - Comparar se a distância entre as coordenadas e a geometria da célula é igual a 0 (0.000001)

Este foi um dos módulos mais complexos de se desenvolver uma vez que teve de ser feito um grande processamento da informação geospacial raster. Para além das verificações normais (projeções), a transformação para GPKG e a interseção com as células da Grelha de Referência fez com que se sentissem bastantes dificuldades. Contudo, os desafios foram ultrapassados com a aplicação de estratégias para contornar a precisão de coordenadas e com as funcionalidades do GDAL para processamento e transcrição da informação raster.

5.1.10 GeoVis

A aplicação interna GeoVis é uma abreviação para Geo-Visualização. É uma aplicação constituída por três módulos: o módulo de Gestão de Gráficos, o módulo de Gestão de Camadas e o módulo de Gestão de Legendas. Estes endpoints, apesar de não estarem diretamente envolvidos com dados geospaciais têm alguma relação como iremos ver na descrição de cada um dos módulos.

Gestão de Gráficos

O módulo de Gestão de Gráficos, constituído por apenas nove endpoints, está relacionado com a representação de dados geospaciais em gráficos. Os gráficos podem ser de vários tipos (barras, scatter, linhas, entre outros).

Para adicionar um novo gráfico geospacial ao sistema foi necessário ter em conta as séries dos gráficos. As séries, no contexto do sistema FireLoc, dizem respeito às cores das barras, pontos ou linhas, dependendo do tipo de gráfico em questão. É por isso que um gráfico estará associado a uma ou mais séries. As séries por sua vez estão associadas aos dados de um determinado gráfico, implicando relacionar vários modelos da base de dados o que acabou por tornar complexo o processo de desenvolvimento destes endpoints, devido à obrigatoriedade de verificação de associações e manipulação de vários modelos.

Gestão de Camadas

O módulo de Gestão de Camadas, constituído por apenas seis endpoints, está relacionado com a referência dos serviços disponíveis do GeoServer. É importante registar que serviços gerados pelo GeoServer se encontram disponíveis, de forma que sejam pedidos à medida que forem sendo necessários.

Este módulo não deve ser confundido com os módulos do SDI. Os módulos do SDI dizem respeito a outra aplicação interna que será descrita mais em frente. Este módulo limita-se a ter referências que permitam associar a informação entre o sistema FireLoc e os serviços gerados pelo GeoServer.

Gestão de Legendas

O módulo de Gestão de Legendas, constituído por seis endpoints, está relacionado à descrição das camadas geospaciais apresentadas no módulo anterior. As legendas clarificam o significado de uma determinada camada para que a análise e representação da informação seja explícita.

O desenvolvimento dos endpoints tiveram em consideração o modelo das camadas uma vez que estão diretamente relacionadas. Apesar desta relação, o desenvolvimento dos endpoints não apresentaram qualquer dificuldade, sendo que as técnicas utilizadas para a sua implementação já foram descritas em módulos anteriores.

5.1.11 Meteo

A aplicação Meteo é uma abreviação para Meteorologia. É uma aplicação constituída por cinco módulos: o módulo de Gestão de Previsões Meteorológicas, o módulo de Gestão de Observações Meteorológicas, o módulo de Gestão de Fontes Meteorológicas, o módulo de Gestão de Estações Meteorológicas e o módulo de Gestão de Variáveis Meteorológicas. Alguns destes módulos, estão relacionados com informação geospacial como iremos ver de seguida.

Antes de descrevermos os módulos que os constituem, é importante destacar que toda a informação meteorológica é de origem externa, mais precisamente da API do IPMA. A informação meteorológica é uma informação difícil de gerir uma vez que é necessário estar, constantemente, a atualizar a informação. Para automatizar o processo, desenvolveu-se Cron Job Services para aceder à API do IPMA todos os dias. Os Cron Jobs permitem automatizar comandos ou scripts específicos para realizar tarefas periódicas automaticamente, e são muito úteis para o sistema FireLoc, pois um Cron Job pode ser configurado para ser executado de 15 minutos em 15 minutos ou de hora em hora, um dia da semana ou mês, ou qualquer combinação destes, o que permite personalizar os momentos em que um determinado script deve executar. [8].

Gestão de Previsões Meteorológicas

O módulo de Gestão de Previsões Meteorológicas é constituído por seis endpoints e, como foi explicado, a sua informação parte da API do IPMA. Para ser mais preciso, está a ser extraída a informação da “Previsão Meteorológica Diária até 3 dias, informação agregada por dia” e que pode ser encontrada em [44].

Os cron jobs apenas acedem a esta API e guardam a informação em diretorias, havendo necessidade de desenvolver um script para processar os ficheiros com as previsões meteorológicas e, por sua vez, realizar as devidas operações de escrita na base de dados.

Para adicionar as previsões meteorológicas ao sistema teve-se de ter em consideração alguns aspetos, nomeadamente, as variáveis meteorológicas relacionadas com as previsões. Por esta razão, podemos assumir que o módulo de Gestão de Variáveis Meteorológicas é complementar e, apenas, se limita a registar variáveis como a velocidade do vento, humidade, entre outros, assim como os valores previstos. Estas previsões, como é natural, estão associadas a uma dada localidade o que implica que haja algum processamento de informação geospacial, nomeadamente, da geometria associada à previsão.

Gestão de Observações Meteorológicas

O módulo de Gestão de Observações Meteorológicas é constituído por seis endpoints e, também, pela sua informação parte da API do IPMA. Para ser mais preciso, está a ser extraída a informação da “Observação Meteorológica de Estações (dados horários, últimas 24 horas)” e que pode ser encontrada em [44].

As observações, como foi revelado, são observações relativamente a estações meteorológicas, inferindo que o módulo de Gestão de Estações Meteorológicas é, também, um módulo complementar. As observações têm, também, variáveis meteorológicas associadas, o que implica ter em consideração vários modelos. Tal como no módulo anterior, foi desenvolvido um script para a limpeza e extração de dados relativamente às observações.

Gestão de Fontes Meteorológicas

Em relação à Gestão de Fontes Meteorológicas, este módulo foi desenvolvido para controlar as APIs que estão a ser utilizadas para extrair a informação, isto é, o sistema FireLoc regista as APIs e associa os módulos que têm sido descritos até agora com a sua fonte, de forma que se possa comparar previsões e observações de outras fontes externas (APIs).

Gestão de Estações Meteorológicas

As estações são, também, extraídas de APIs externas o que implicou, novamente, implementar um script para a extração dos dados. Este módulo, constituído por seis endpoints, limitou-se a registar a localização das estações e associar estas estações com as fontes de forma a controlar toda a informação da aplicação Meteo.

Gestão de Variáveis Meteorológicas

Por último, o módulo de Gestão de Variáveis Meteorológicas serviu de complemento para gerir as variáveis relacionadas com as previsões e observações meteorológicas. Estas variáveis permitem registar os valores das previsões e observações associadas.

A aplicação Meteo baseou-se em cron jobs, scripts de extração e processamento de informação geospacial. Os módulos apresentados implicaram realizar várias atividades complementares para se conseguir ter a informação no sistema FireLoc. Houve algum nível de complexidade relativamente ao desenvolvimento de scripts, uma vez que a estrutura dos ficheiros JSON das previsões, observações e estações eram distintos entre si, havendo a necessidade de adaptar os scripts e realizar um processamento diferente para cada um dos módulos.

5.1.12 SDI

A aplicação SDI é especial, uma vez que consiste na realização de operações num servidor externo, o GeoServer. Para clarificar em que consiste esta aplicação, vale recordar a figura 2.16 do Capítulo 2 na secção 2.5.1. Nesta figura é representado como é feita a interação entre o cliente, a API FireLoc e o GeoServer para gerar serviços de mapas, sendo o principal foco desta aplicação. Os utilizadores finais vão realizar sempre os pedidos à API FireLoc e, os módulos que irão ser descritos de seguida, serão os responsáveis por reencaminhar

os pedidos para o servidor externo (GeoServer). O GeoServer, por sua vez, devolve uma resposta com os serviços gerados pelo mesmo e que serão consumidos pelo Leaflet para renderizar a informação no mapa.

Para gerir todo este fluxo entre o cliente e os servidores, foram desenvolvidos quatro módulos, estando três deles relacionados à forma como o GeoServer gere a informação e que já tem sido abordado em mais de uma ocasião: Workspaces, Layers e Stores. Por essa razão, cada um destes conceitos deve ter um módulo associado, de forma que, o sistema FireLoc consiga interagir com o GeoServer e realizar as operações de forma controlada e organizada. O quarto módulo é um dos mais importantes por ser o responsável pela obtenção dos serviços de mapas.

Gestão de Workspaces - GeoServer

O módulo de Gestão de Workspaces é, como o próprio nome indica, um módulo destinado para realizar operações CRUD em relação aos workspaces do GeoServer, o qual é constituído por quatro endpoints.

Os endpoints desenvolvidos consistem na realização de pedidos ao GeoServer e, foram construídos baseados na documentação do Swagger do GeoServer que se pode encontrar em [34]. É importante referir que os restantes módulos desta aplicação interna estão diretamente associados à Gestão de Workspaces.

Gestão de Stores - GeoServer

O módulo de Gestão de Stores segue um processo semelhante ao módulo anterior. É constituído, também, por quatro endpoints e estão baseados na documentação do GeoServer em [33].

Para adicionar uma nova store ao GeoServer e, comparando com adição de workspaces, o processo é diferente. As stores podem ser de vários tipos, no contexto do sistema FireLoc estamos a considerar dois tipos: Stores PostGIS e Stores Raster. Deste modo, existem dois métodos para a criação de Stores, um específico para cada tipo.

Em cada um dos tipos, foi criado um ficheiro Extensible Markup Language (XML) temporário para a configuração das stores, pois é um dos requisitos para o processo de criação, segundo a documentação do GeoServer. Na configuração é especificado o tipo de store, o nome, o workspace, entre outras informações. Estes ficheiros de configuração estão a ser construídos e guardados na API FireLoc.

O ficheiro de configuração XML é enviado ao GeoServer no pedido REST. É de notar que uma store tem de, obrigatoriamente, estar associada a um determinado workspace e, portanto, deve ser especificado no pedido. O processo para criar stores raster segue a mesma estratégia, adaptando o ficheiro XML, no qual deve ser especificada a localização no sistema de ficheiros que contém os dados raster (GPKG). Uma vez que as stores são criadas com sucesso, os ficheiros de configuração XML são apagados do sistema.

Gestão de Layers - GeoServer

O módulo de Gestão de Layers depende, obrigatoriamente, da existência de um workspace e de uma store. A store é um fator muito importante, pois define quais dados é que podem

ser usados para gerar os serviços de mapas. Quando é desejado adicionar uma camada (layer) ao GeoServer está só poderá ser adicionada a partir da store, isto significa que não podemos adicionar camadas que não se encontrem nas stores.

O processo para a criação de layers é semelhante ao módulo de stores em relação ao ficheiro de configuração XML. Contudo, é necessário, como passo adicional, especificar a tabela PostGIS (camada) que se pretende publicar. Caso a tabela não faça sentido em termos geográficos, isto é, publicar uma tabela sem contexto geospacial, como, por exemplo, a tabela onde estão registados os utilizadores, o GeoServer publica a tabela mas não será reconhecida nos pedidos da API FireLoc. O GeoServer lida com qualquer tipo de informação seja ela geospacial ou não, mas em termos de representação, como é natural, só consegue exibir informação geospacial nos mapas (serviços gerados).

Obtenção de Serviços - GeoServer

Este módulo é o responsável pela obtenção dos serviços de mapas do GeoServer que têm sido, frequentemente, abordados nesta dissertação. É um módulo muito importante do ponto de vista funcional. Apesar de ser, apenas, constituído por dois endpoints, implicou que fosse feita uma análise atenta de todos os parâmetros que o GeoServer necessita para obter um serviço WMS, por exemplo. Este módulo está diretamente relacionado com o módulo GeoVis apresentado na secção 5.1.10. Apenas podem ser obtidos os serviços que estão registados na aplicação GeoVis em relação às camadas.

O processo que o GeoServer efetua é invisível para os programadores e muito mais para os utilizadores finais. Cada parâmetro especifica uma camada de abstração no pedido REST. Entre os parâmetros mais importantes, podemos destacar o serviço (WMS ou WFS), as layers e o formato. Dependendo do serviço, o formato poderá variar de PNG para JSON, por exemplo. É importante destacar que os endpoints invocados pelos utilizadores finais devem ser personalizados com os parâmetros conforme o pedido.

5.1.13 Módulo de Recolha e Harmonização dos dados

Este é um módulo que não faz parte da API propriamente dita, como se pode observar nos modelos da Arquitetura do Sistema, no Capítulo 4. Contudo, tem sido mencionado, de forma implícita, em vários módulos desta secção. É constituído por todos os scripts descritos, nomeadamente, na recolha de dados de APIs externas, como o Catálogo Geográfico do ICNF, a API do IPMA, o Portal de dados abertos da Administração Pública fornecidos pelo Governo de Portugal, imagens de satélite, entre outros.

A recolha destes dados e a sua harmonização são, na prática, o desenvolvimento destes scripts e que permitiram ao sistema FireLoc obter toda esta informação. Apesar de não fazer parte da API, foram realizadas atividades em paralelo com API por estarem relacionadas em certos módulos e, portanto, merece ser destacado nesta secção.

5.2 Desenvolvimento com o GeoServer

O desenvolvimento com o GeoServer baseou-se, em grande parte, no desenvolvimento da API, isto é, ambas são dependentes uma da outra dado que o objetivo é utilizar os serviços REST do GeoServer a partir da API FireLoc. Nesta secção, são abordadas as atividades relacionadas com o GeoServer e a conceção por trás dos serviços.

Em termos de utilização, a vertente de front-end nunca comunica diretamente com o GeoServer, isto significa que o lado de cliente nunca irá interagir diretamente com o servidor geospacial, sendo esse um dos motivos para o isolar. Por outro lado, a API FireLoc serve de ponte para comunicar com os serviços REST do GeoServer, como é ilustrado na figura 2.16 e, desta forma, conseguir aceder aos seus serviços de web mapping.

Dito isto, o único servidor que esta exposto para o exterior é o do FireLoc, que inclui a API. Uma outra razão para se isolar o GeoServer são questões de segurança, o que permite evitar configurações no GeoServer e preocupações com outro sistema de gestão de utilizadores (acesso com permissões). Desta forma, garantimos que todas as tentativas de acesso feitos ao GeoServer passam, inicialmente, pelo sistema de autorização da API.

5.2.1 Criação de Serviços - Mapas

O cerne da integração do GeoServer no projeto é baseada na criação de serviços, ou seja, mapas a serem consumidos pela aplicação cliente. Nestas atividades em particular apenas se abordaram os dados geospaciais vetoriais por serem, neste momento, mais importantes no projeto. A informação que se irá manipular está diretamente relacionada com os incêndios florestais, nomeadamente, as contribuições dos utilizadores, eventos reais e eventos identificados pelo sistema.

A criação de serviços é realizada no módulo de Gestão de Camadas da aplicação GeoVis. Para criar um serviço devem ser especificadas uma série de parâmetros, nomeadamente, os workspaces, stores e, evidentemente, as camadas (dados geospaciais), como foi revelado na descrição do GeoVis. Estes serviços por sua vez irão ser invocados através do módulo da Obtenção de Serviços do GeoServer na aplicação SDI.

Renderização dos Serviços - Leaflet

Apesar do front-end não ter sido registado como uma das tarefas nesta dissertação, para que se pudessem testar os serviços do GeoServer houve a necessidade de criar uma simples aplicação web de cliente, com um mapa (Leaflet) que se limita a fazer pedidos à API FireLoc para obter os serviços de mapas gerados pelo GeoServer. O objetivo da criação deste mockup, é observar o comportamento do Leaflet e medir a renderização dos dados geospaciais, assim como identificar algumas técnicas e/ou estratégias de melhoria de performance de renderização. No Capítulo 6 são descritos mais detalhes, nomeadamente os testes realizados assim como os seus resultados.

5.3 Desenvolvimento com Docker

Como foi descrito no Capítulo 2, na secção das Tecnologias, o sistema FireLoc está a usar tecnologias de virtualização para containerizar e independizar o software desenvolvido. No contexto do sistema, é importante termos micro-serviços e aplicações prontas a executar. Assim sendo, nesta secção serão descritas as atividades realizadas de desenvolvimento que implicaram usar o Docker como principal tecnologia de desenvolvimento.

5.3.1 Cron Job Services

Para que o sistema FireLoc consiga automatizar alguns processos, como a execução do módulo apresentado na secção 5.1.13 (Recolha e Harmonização de dados) ou obter as novas contribuições, foi desenvolvido um serviço em Docker para a execução de cron jobs. Os cron jobs foram descritos de forma sucinta na secção 5.1.11, no entanto, será descrito de forma mais detalhada em que consiste e como podem ser usados.

Cron Jobs

Um cron job é, na prática, um comando do Linux usado para agendar tarefas a serem executadas em algum momento no futuro [42], podendo ser qualquer tipo de script executável. Como se está a usar Python no projeto, os scripts estão a ser escritos nesta linguagem de programação.

Para que o cron job seja, efetivamente, executado é necessário realizar a configuração do período em que o mesmo deverá ocorrer, num ficheiro conhecido como Cron Tab. A configuração é feita seguindo uma sintaxe muito simples.

*	*	*	*	*	comando(s)
-	-	-	-	-	
					— Dia da semana (0 - 7)
			-		— Mês (1 - 12)
		-	-		— Dia do mês (1 - 31)
	-	-	-		— Hora (0 - 23)
-	-	-	-		— Minuto (0 - 59)

Tabela 5.4: Sintaxe do Cron Tab

Como se pode observar na tabela 5.4, a sintaxe tem um prefixo com significado temporal seguido de um comando para a execução do script.

Container para Cron Jobs

Para que, de facto, o serviço tenha efeito no sistema FireLoc, foi criada um container em Docker para a execução destas tarefas baseado na aplicação GRASS. A aplicação GRASS é, como já foi mencionado, uma aplicação usada para a gestão de dados geoespaciais e análise, processamento de imagens, gráficos e mapas de produção, modelagem espacial e visualização [79].

O container foi adaptado ao projeto, o que implica a integração de várias dependências do sistema, como a biblioteca Firegis (exclusiva do sistema FireLoc), de forma que, seja possível a interpretação dos módulos FireLoc no container. Para efeitos de teste, foi desenvolvido um script muito simples que consiste em obter as contribuições do sistema. No bloco de código 5.3 está representado o Cron Tab do container criado com a especificação do script a ser executado cada minuto.

```
*/1 * * * * grass PYTHONPATH=/firegis /usr/bin/python3 -u /firegis/firegis/
  services/psql.py >> /var/log/cron.log 2>&1
# Every min run this script
```

Bloco 5.3: Cron tab do sistema FireLoc

Na figura 5.5 pode-se observar o log do container criado, cujas respostas são do cron job descrito.

```
Initializing Cronjob Request at: 16:25:01
Connection to API sucessfully.
HTTP/1.1 200 OK
Cronjob executed successfully.

Initializing Cronjob Request at: 16:26:02
Connection to API sucessfully.
HTTP/1.1 200 OK
Cronjob executed successfully.

Initializing Cronjob Request at: 16:27:01
Connection to API sucessfully.
HTTP/1.1 200 OK
Cronjob executed successfully.
```

Figura 5.5: Log do Container para execução de Cron Jobs.

5.3.2 Aplicações de Catálogo

As aplicações de catálogo são baseadas nas UIs que apresentam para um conjunto de funcionalidades específicas e que, no contexto do sistema FireLoc, consiste em aplicações para a gestão de informação geospacial. No entanto, existem aplicações de catálogo, como foi revelado no Capítulo 2, que habilitam funcionalidades de administração de utilizadores e de grupos, tornando as interfaces num ambiente administrativo. No Capítulo 2, fez-se, também, uma comparação entre aplicações de catálogo que se pudessem integrar no sistema, acabando por escolher-se o GeoNode para a realização de experiências.

Visto que o GeoNode é uma aplicação por si só, faz sentido que a sua integração com o sistema FireLoc seja feita de forma semelhante como ocorreu com o GeoServer e o PostgreSQL, isto é, por meio de virtualização e isolamento da aplicação usando Docker.

Container para GeoNode

A containerização do GeoNode foi realizada seguindo os passos da sua própria documentação e que pode ser encontrada em [30]. A instalação do GeoNode implicou ter todo o repositório da aplicação no sistema e adaptar o Dockerfile, tal como foi feito com os Cron Job Services, de modo a que se conseguisse independizar a aplicação de terceiros. As atividades realizadas com o GeoNode não foram extensas, visto que o objetivo principal era a familiarização com o Docker, explorar o funcionamento da aplicação e identificar as dependências em termos funcionais da aplicação (bibliotecas, módulos, entre outros).

Durante a exploração ficou claro como é que administradores e gestores do FireLoc poderiam utilizar o GeoNode para as suas atividades administrativas recorrendo à extensão de back-office do Django e secções de monitorização e análise estatística, tudo disponibilizado pela aplicação GeoNode.

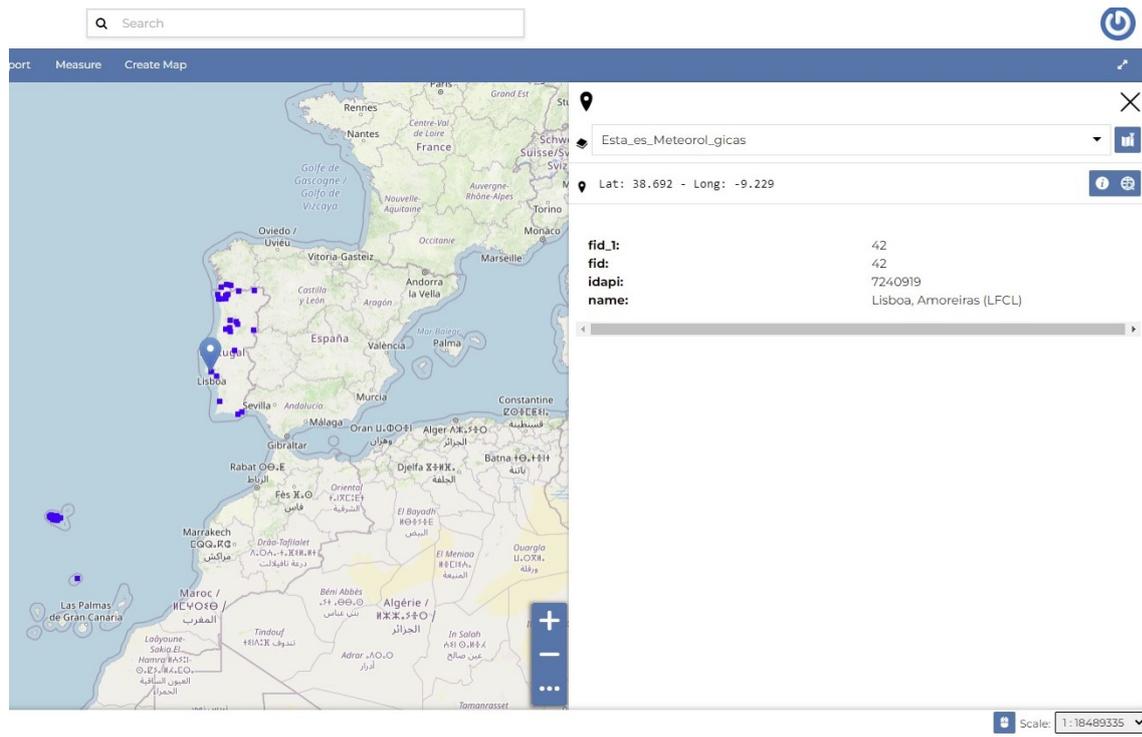


Figura 5.6: Renderização de Estações Meteorológicas no GeoNode

Na figura 5.6 pode-se observar um exemplo da renderização de dados geospaciais no GeoNode, mais especificamente as estações meteorológicas identificadas e geridas pela aplicação interna Meteo (secção 5.1.11). Esta informação foi gerada a partir do GeoServer - integrado no GeoNode.

5.4 Sistema FireLoc em Produção

Realizaram-se algumas atividades para habilitar o acesso público à API do sistema FireLoc, isto é, colocar a API, a base de dados e o GeoServer em produção. Para a realização desta atividade, seguiram-se um conjunto de passos e protocolos obrigatórios para garantir que o sistema é acessível remotamente e trabalhar diretamente nos servidores onde corre o sistema FireLoc. Nesta secção, são descritos todos os passos de configuração e tecnologias complementares usadas para o efeito.

5.4.1 Deployment do Sistema

Para o deployment do sistema, alguns dos passos realizados são idênticos aos que foram feitos para o setup inicial, que o estagiário teve de configurar para as suas atividades locais. O primeiro passo é criar um ambiente virtual e instalar as dependências do sistema FireLoc para que todos os comandos e operações realizadas sejam reconhecidas.

O segundo passo foi instalar e configurar o Apache. O Apache é um servidor web HTTP que permite alocar e correr aplicações. Neste contexto, é a tecnologia utilizada para suportar e correr o servidor FireLoc. O Apache não se limita apenas a suportar o servidor FireLoc mas, também, funcionará como *reverse proxy*. Este modo de funcionamento implica que os clientes comuniquem e façam os pedidos ao Apache e estes por sua vez sejam

reencaminhados para o servidor FireLoc. Para garantir segurança na web, estão a ser usados certificados SSL, como foi revelado na secção 5.1.3. Para gerar os certificados e serem interpretados pelo Apache, foi usado o Certbot.

O Certbot é uma aplicação intermediária que permite gerar certificados SSL através do Let's Encrypt (serviço que distribui certificados digitais gratuitos). Deve ser associado o domínio público utilizado aos certificados digitais para que possa ser habilitado o protocolo HyperText Transfer Protocol Secure (HTTPS), extensão segura do HTTP.

Para além disto, foi necessário construir e executar os containers do sistema, nomeadamente, o container da API, do GeoServer e da base de dados. Todos estes containers foram configurados para que sejam interpretados pelo Apache e sejam compatíveis com o protocolo HTTPS através do domínio público, propriedade do sistema FireLoc.

5.4.2 Desafios Associados

Colocar o sistema FireLoc em produção revelou ser um processo bastante complexo. Apesar de muitos passos serem idênticos à configuração local, enfrentaram-se vários problemas para colocar os containers disponíveis e acessíveis ao público. Entre os problemas, pode-se destacar a migração dos modelos e população da base de dados, assim como a execução de alguns containers. Na verdade, muitos dos problemas desencadearam outros, uma vez que todos estes conceitos e atividades estão diretamente relacionados. Problemas com a execução do container da base de dados impediu que alguns processos de população da base de dados fossem comprometidos. Estes problemas não eram óbvios, uma vez que os Logs dos containers não indicavam, exatamente, quais eram os principais erros. Contudo, chegou-se à conclusão de que alguns scripts estavam a subestimar o tempo de construção e execução dos containers, o que implicou lidar com problemas assíncronos entre os containers e scripts. A solução para este problema foi aumentar a tolerância de tempo que o sistema FireLoc permite para a construção e execução dos containers.

Relativamente aos problemas na população da base dados, criou-se um script para a população através de um ficheiro SQL respeitando os modelos existentes atuais do Django.

Gerar este ficheiro, implicou a utilização da aplicação externa DB Backup descrita na secção 5.1.1 com comandos próprios da aplicação. Estes processos não foram simples pelo facto de haver constantes mudanças na estrutura dos modelos e correção de alguns módulos, pelo que a restauração da base de dados tornou-se uma atividade comum.

Restaurar a base de dados, por sua vez, implica que sejam reconstruídos os containers para que corram com a nova estrutura e alguns módulos não sejam comprometidos. Todo o processo descrito foi automatizado com scripts em Python e com Docker Compose, contudo, apesar da automatização, o processo apresentou diversas barreiras, tendo sido uma das atividades mais complexas em termos de trabalho.

Capítulo 6

Validação do Sistema

Na maioria dos projetos de software, os testes são uma atividade final, porém, se o planejamento para os testes for paralelo ao desenvolvimento dos requisitos, muitos erros podem ser encontrados logo após a sua introdução, evitando, assim, que causem mais “danos” no processo de desenvolvimento. Visto que, o sistema FireLoc não se encontra completamente integrado não poderemos realizar testes no sistema como um todo, pois o software não está pronto para ser exposto ao público em geral. No entanto, podem ser realizados outros tipos de testes que nos permitem validar partes do sistema.

Dito isto, será comprovado o bom funcionamento dos módulos desenvolvidos, assim como certificar que os objetivos foram alcançados com sucesso. Os testes realizados confirmaram os estudos feitos em 1996 por Blackburn, Scudder e Van Wassenhove em relação à revisão e testes de requisitos, no que concerne a encurtar o tempo total de desenvolvimento de um sistema [46].

São apresentados todos os testes relacionados com a implementação do sistema, nomeadamente, para a validação e verificação de RFs (Testes Funcionais) e dos atributos de qualidade identificados (Testes Não Funcionais).

6.1 Testes da API

Como foi mencionado no Capítulo 3, a Validação é a quarta e última fase do ciclo de vida clássico de desenvolvimento de requisitos. No entanto, a validação não pode ser considerada sem ocorrer uma Fase de Verificação. A fase verificação determina se uma solução cumpre com os requisitos, isto é, se aplica todos os conceitos, boas práticas e procedimentos adequados. A validação, por outro lado, permite-nos determinar se a solução desenvolvida é correta e reflete o uso esperado de software.

6.1.1 Fase de Verificação

Visto que esta dissertação está enquadrada num estágio supervisionado, o coorientador realizou as atividades de verificação das funcionalidades desenvolvidas. O processo de revisão foi informal, no qual o coorientador identificava pontos a melhorar em termos de funcionamento e estrutura de código. A revisão era, posteriormente, submetida e discutida em reuniões semanais de forma a que se pudessem aplicar as melhorias aos módulos e/ou componentes.

6.1.2 Fase de Validação

Todas as funcionalidades desenvolvidas passaram, pelo menos, por duas fases de validação: uma antes da fase de verificação por parte do coorientador e outra após se realizar uma discussão da revisão. Todos os testes de validação foram feitos em paralelo com o desenvolvimento dos componentes. Na figura seguinte está representado o ciclo de testes ao qual cada funcionalidade esteve sujeita.

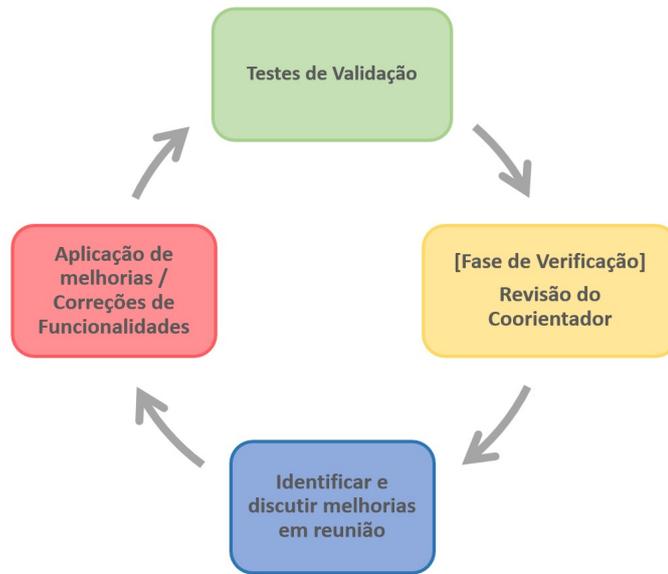


Figura 6.1: Ciclo de Testes

6.1.3 Testes Unitários

Os testes de validação realizados tiveram em conta o bom funcionamento dos endpoints e por isso, foram feitos testes unitários. Os testes unitários são testes individuais das unidades que constituem uma classe ou módulo. Neste caso em específico, pode-se assumir que os módulos são, precisamente, as aplicações internas desenvolvidas e descritas no Capítulo 5, e as unidades, cada um dos endpoints que o constituem.

Considerando os Critérios de Aceitação definidos no Capítulo 3 para cada um dos requisitos, pode-se assumir que estes critérios são os oráculos para o teste em questão. Para relembrar, os critérios de aceitação definem as condições mínimas para que uma unidade possa ser validada e fechar o ciclo de vida do requisito a que está sujeito.

6.1.4 Testagem com Jupyter Lab - Notebooks

Para a realização dos testes, foram construídos notebooks na plataforma Jupyter Lab. O Jupyter Lab funciona como um ambiente de execução em Python o que permite construir testes rapidamente. Os notebooks são ficheiros particulares do Jupyter Lab e que se encontram no projeto para que, qualquer membro da equipa FireLoc, consiga, efetivamente, testar a API. Desta forma, os notebooks funcionam como uma espécie de documentação, na qual é descrita a utilidade de cada endpoint e uma breve descrição do módulo. Contudo, os notebooks precisam de ser escritos pelos programadores e implica que seja despendido

algum tempo e esforço para a sua construção, tanto para a documentação como para os testes. Para além disso, durante a testagem exaustiva dos endpoints de cada módulo, houve a necessidade de automatizar o processo. Porém, para que a automatização incluísse os critérios de aceitação (oráculos) era necessário desenvolver uma série de scripts complexos. Por esse motivo, os testes unitários passaram a ser realizados na aplicação Postman, cuja automatização era uma das suas principais funcionalidades e evitar “reinventar a roda”.

6.1.5 Testagem com Postman - Swagger

O Postman é uma aplicação desenhada para programadores com o objetivo de facilitar a construção e a testagem de APIs. Possui um grande conjunto de ferramentas para esse fim, reduzindo o tempo e esforço despendido na fase de testes. Essa foi uma das principais razões pela qual se optou por realizar os testes unitários no Postman e não com os notebooks. Apesar desta decisão, os notebooks não foram descartados e estão disponíveis para membros da equipa que não optem por utilizar software complementar para a testagem. Para além disso, os notebooks têm os scripts desenvolvidos para a extração de dados de fontes externas e, portanto, são indispensáveis. O Postman está a ser utilizado exclusivamente para testes.

Por outro lado, também se utilizou o OpenAPI/Swagger. O Swagger é uma plataforma que permite gerar, automaticamente, um documento web informativo com base no código escrito, isto é, os endpoints da API FireLoc. O documento especifica todo o funcionamento e estrutura dos endpoints, bem como realizar chamadas à API, fazendo dela uma espécie de documentação de todos os endpoints existentes. O Swagger é completamente integrável com o Postman e, desta forma, pode ser migrado todo o documento gerado pelo Swagger com o objetivo de criar todos os endpoints da API automaticamente.

Procedimento

Os critérios de aceitação definidos ditaram o procedimento para a realização dos testes no Postman. Neste sentido, foi criado um *test-suite* (conjunto de testes para validar determinados comportamentos) para a validação das unidades testadas, sendo constituído pelas seguintes métricas:

- Validação do Status Code;
- Validação da Estrutura do Serializer do Modelo.

O critério de aceitação em relação a garantia de sucesso, isto é, da funcionalidade propriamente dita, terá de ser validada no momento da sua utilização. Por outro lado, o critério de rejeição (tempo médio de resposta superior a 1 segundo) será observado em cada execução.

```
pm.test("Validating Status Code", () {
  pm.response.to.have.status(200); //Depends on HTTP method
});

pm.test("Validating Serializer Structure", () => {
  pm.expect(pm.response.json()).to.be
    .jsonSchema(srl_structure) //Depends on Model
});
```

Bloco 6.1: Test-Suite - Exemplo

No bloco de código 6.1 é representado o exemplo de um *test-suite* na qual é esperado uma resposta HTTP 200 (Dados devolvidos com sucesso) que respeite a estrutura do serializer definido num objeto JSON construído individualmente e adaptado para cada unidade (endpoint).

Automatização de Testes

A automatização dos testes da API é algo obrigatório para que os resultados e a análise sejam mais apurados. Não só facilita a realização dos testes, como torna o processo mais rápido e correto no contexto de engenharia.

Desta forma, a automatização dos testes no Postman foi realizada, de forma que a execução dos testes se repetisse cem vezes, sendo assim uma automatização iterativa. Em cada uma das iterações são aplicados os test-suites, tendo cada teste um oráculo associado. Os oráculos são os seguintes: Passou todos os testes; ou Falhou um dos testes.

No fim dos testes, é gerado um relatório sumariado indicando o tempo de resposta do pedido à API em cada uma das iterações e o resultado de cada iteração (resultado do oráculo). Na figura 6.2 pode-se observar o relatório gerado na qual é especificado o resultado de cada iteração, assim como os tempos de resposta. Este processo foi repetido em todos os endpoints desenvolvidos na API.

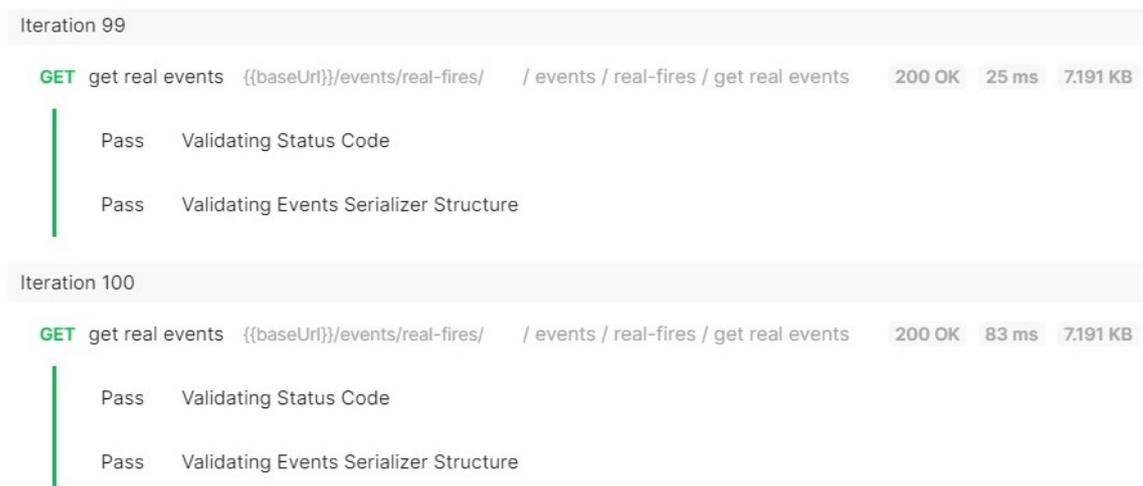


Figura 6.2: Resultados da Automatização de Testes - Exemplo

Resultados

Para a representação dos resultados, construiu-se um gráfico (Boxplot) para cada um dos métodos HTTP. Desta forma, os gráficos são construídos em função dos módulos da API a fim de representar a performance em termos de tempo de resposta. Para que a comparação entre tempos de resposta seja justa e imparcial, método HTTP tem de ser o mesmo, assim como o número de registos envolvidos no pedido, fixado em cem (100).

Os endpoints complementares não foram considerados na representação dos gráficos, ou seja, todos os endpoints relacionados com atributos e/ou componentes que existam apenas como associação a outros, nomeadamente, a Autenticação, Grupos, Atributos de Utilizadores, Anos dos Eventos, Abordagens, Atributos de Localização de Incêndio, Atributos de

Classificação de Fotografias, Módulo GeoRef, Tipos Raster, Series, Variáveis Meteorológicas, e Fontes Meteorológicas. Os fatores mais importantes e que irão ser analisados são os seguintes:

- Ultrapassou os Critérios de Aceitação (Testes);
- Média do tempo de resposta (Critério de Rejeição);
- Ter ou não informação geospacial envolvida.

Na figura 6.3 podemos observar a representação dos resultados obtidos em relação aos pedidos GET em termos de listagem, isto é, em obter todos os registos de cada um dos módulos. O gráfico em relação ao método POST encontra-se no Apêndice D. Os métodos GET (específico), PUT e DELETE (todos e específicos) foram omitidos.

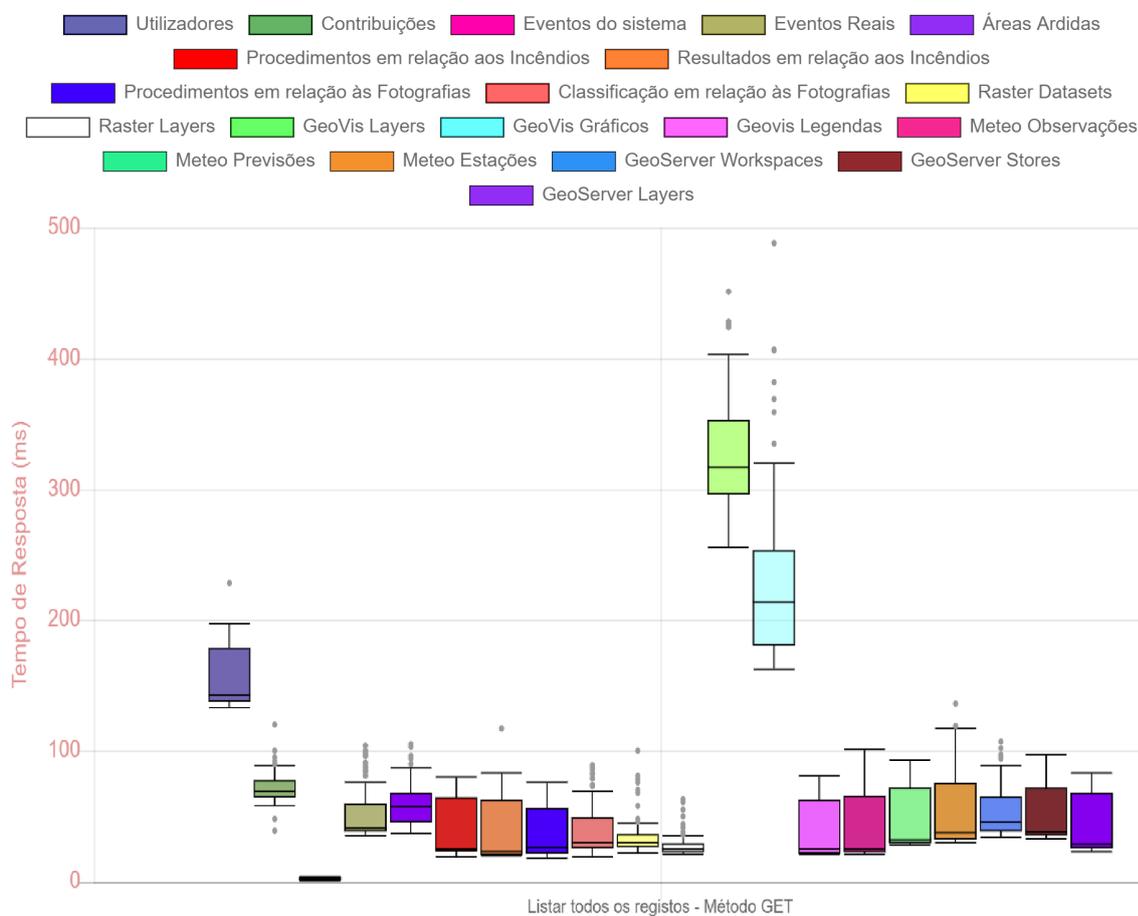


Figura 6.3: Resultados para o método GET (listagem) - Boxplot

Discussão

Na análise dos resultados (gráficos), mais precisamente na figura 6.3, pode-se destacar três componentes com elevados tempos de resposta em comparação com os restantes módulos, nomeadamente, o módulo de Utilizadores, Gráficos geospaciais e Camadas geospaciais. Estes módulos apresentam uma média de 156.28 *ms*, 326.02 *ms* e 228.95 *ms*, respetivamente, sendo desproporcional à média global de 72.74 *ms*.

Cabe destacar que as médias não se encontram representadas nos gráficos e, portanto, para chegar aos valores apresentados houve a necessidade de fazer os cálculos à parte. Estes tipos de gráficos são úteis para representar e analisar a variação de um conjunto de dados.

Todos os testes cumpriram com os critérios de aceitação definidos, nomeadamente, a resposta do pedido HTTP e a estrutura da resposta em relação ao serializer. O tempo de resposta como se pode observar não ultrapassou os 500 *ms*, isto significa que o critério de rejeição nunca foi presenciado nos testes, indicando que o tempo médio de resposta está dentro dos valores esperados.

Conclusão

Os resultados obtidos demonstram que, para o método GET, relativamente a listagem de todos os registos, não houve qualquer relação entre ter ou não informação geospacial envolvida. Por outro lado, a construção da resposta feita no desenvolvimento dos módulos tem uma grande influência no tempo de resposta dos pedidos.

Os módulos com uma média superior à média global de performance (Utilizadores, Camadas geospaciais e Gráficos geospaciais) têm, no conteúdo das suas respostas, mais do que um modelo associado. Esta relação entre modelos na resposta obriga a que os pedidos sejam mais demorados uma vez que as consultas à base de dados têm várias tabelas associadas.

O gráfico apresentado para os utilizadores foi o único que não respeitou a igualdade no número de registos devolvidos no pedido. Como foi mencionado, o número de registos fixou-se em cem, para que houvesse imparcialidade na comparação. Contudo, o número de registos para os utilizadores, neste exemplo, é de seis registos. O motivo pelo qual se fixou em seis registos para este componente, foi o facto do tempo de resposta ser completamente arrasador, em termos de superioridade, comparativamente aos restantes componentes. Consequentemente, a representação do gráfico prejudicava os restantes em termos de visualização (visualização desproporcional) pelo que se reduziu drasticamente o número de registos. O facto do componente de Utilizadores ser tão “pesado” é pelo facto de ter quatro modelos envolvidos, os grupos, os atributos de utilizador, os valores dos atributos, e os tokens associados ao utilizador o que obriga a um processamento demorado da resposta.

Cabe destacar que, apesar de alguns componentes terem elevados tempos médios de resposta, a média está dentro do intervalo (inferior a 1 segundo), porém, o tempo de resposta, naturalmente, será afetado pelo número de registos devolvidos. Uma das alternativas para melhorar o tempo médio de resposta será substituir a estratégia do uso de serializers e realizar as consultas à base de dados manualmente, através da construção de queries. Os serializers carregam inúmeras vantagens, entre elas a validação da estrutura dos modelos e dos dados, contudo, é um processo bastante custoso. Esta alternativa será mais complexa em termos de desenvolvimento, porém, poderá ser vantajosa para módulos cujo tempo médio de resposta o exija. Na verdade, esta técnica já está a ser aplicada na Gestão de Contribuições, pois é esperado que seja um dos módulos com mais registos envolvidos, como retratado no bloco de código 5.2.

A realização dos testes exaustivos à API identificaram alguns defeitos no desenvolvimento, nomeadamente, nas respostas obtidas. Estes erros foram, por sua vez, resolvidos, implicando que alguns componentes tenham passado por vários ciclos de teste (6.1). Após a correção dos problemas identificados, todos os endpoints passaram nos testes aplicados (critérios de aceitação). Apesar de terem sido ultrapassados com sucesso, não podemos descartar a existência de defeitos no código.

6.2 Testes de Integração

Atualmente, a equipa de desenvolvimento, responsabilizada por implementar o front-end das aplicações, está a realizar testes de integração das funcionalidades que se encontram completamente validadas (revistas, testadas e aprovadas). Infelizmente, os testes de integração não estão incluídas como objetivos de trabalho. Contudo, foram detetados alguns problemas relacionados com a funcionalidade de alguns componentes pela equipa de front-end.

Entre às falhas de funcionalidade, podemos destacar a violação de segurança encontrada no registo de utilizadores. A equipa de front-end, durante o processo de integração, detetou que os novos utilizadores registados não eram capazes de se autenticar no sistema. Este problema foi notificado de forma que se pudesse detetar e expor a origem do problema.

Através de testes unitários e revisão dos dados na base de dados, verificou-se que as passwords enviadas no registo estavam a ser guardadas em plaintext e não de forma encriptada/hash. Parte do sistema de autenticação do sistema FireLoc é controlado pelo Django, que usa o algoritmo PBKDF2 com um hash SHA256 para manter as passwords seguras e ilegíveis. Esta técnica é bastante segura, exigindo grandes quantidades de tempo de computação para obter a password [22]. Deste modo, houve a necessidade de aplicar um método nativo do Django que permite aplicar o hash nas passwords. Um utilizador só é autenticado após validar e comparar a igualdade entre a password inserida e a password da base de dados.

Dito isto, apesar de não ter havido atividades relacionadas com testes de integração das funcionalidades, houve a preocupação de corrigir os defeitos detetados e realizar novos testes unitários para a validação das funcionalidades.

6.3 Testes de Atributos de Qualidade

Os atributos de qualidade são tão importantes como os requisitos funcionais e devem, também, ser testados para observar a sua existência no sistema. Como foi mencionado, testar a qualidade de uma aplicação é uma temática complexa e, por essa razão, tentou-se que as métricas, definidas para a avaliação, fossem o mais simples possíveis mas, que expusessem a qualidade do sistema sem contestação.

Os testes para os atributos de qualidade são baseados nos cenários apresentados na secção 3.3. Replicaram-se as condições de cada um deles para validar a existência dos atributos a partir da medição da resposta, apresentada em cada um dos cenários.

Para a realização dos testes, seguiu-se a priorização realizada na secção 3.3.5, isto é, os testes foram efetuados do mais prioritário ao menos prioritário. Esta abordagem permitiu que fosse dado privilégio aos atributos com maior pontuação de prioridade o que, por sua vez, possibilitou que fossem abordados com mais antecedência e procurar soluções que os garantissem com urgência por serem considerados os mais importantes para o sistema FireLoc. Na fundamentação da priorização revelaram-se algumas das técnicas e tecnologias que estão relacionadas com estes testes.

Nesta secção serão apresentados todos os testes elaborados em relação aos atributos de qualidade identificados no sistema FireLoc, no que diz respeito ao servidor.

6.3.1 Testes de Segurança no Sistema

A segurança foi considerada como o atributo mais importante para o sistema FireLoc. Por razões evidentes, a segurança é um atributo imprescindível para qualquer sistema, sendo dividida em duas propriedades: Confidencialidade e Integridade. Por se tratar de um projeto web based, o termo segurança deve estar relacionado com as ameaças na web.

O sistema FireLoc confia nos recursos de segurança do Django, tendo uma base robusta contra ameaças baseadas nas indicações da Open Web Application Security Project (OWASP). A OWASP é uma comunidade que disponibiliza documentação e técnicas para proteger aplicações web. As suas indicações baseiam-se em ameaças como Cross-site scripting e SQL Injection, mas não só. O Django aplica, na sua infraestrutura, técnicas que protegem as aplicações, embora cabe, também, aos programadores aplicar algumas estratégias para garantir segurança no sistema.

Testes de Integridade

As ameaças à integridade de um sistema web based são relativas a modificação dos dados dos clientes ou do servidor. As medidas para o tratamento deste tipo de ameaça são, maioritariamente, algoritmos de criptografia. Contudo, o sistema FireLoc baseia-se nas propriedades ACID do PostgreSQL para garantir a integridade. Para a testar, executaram-se inúmeros pedidos à API (consultas à base de dados). As respostas obtidas coincidiram sempre com aquilo que era esperado. Os testes realizados na secção 6.1 vieram também confirmar a integridade do sistema uma vez que foi provado que as centenas de testes processados passaram os critérios definidos, sem alteração da estrutura da resposta nem do seu conteúdo (integridade dos dados) assim como respeitar o propósito da sua construção (integridade do sistema).

Desta forma, pode-se assumir que a integridade no sistema FireLoc é garantida, uma vez que os dados cumprem com a estrutura e conteúdo original. Ainda, a funcionalidade efetua as operações como esperado.

Testes de Confidencialidade

Por outro lado, as ameaças à confidencialidade são ameaças relativas a privacidade. Já foi revelado que o sistema FireLoc utiliza a aplicação OAuth2 como base para o sistema de autenticação e autorização. Estes conceitos são muito importantes para a confidencialidade de um sistema e serão abordados neste tópico.

Testar a confidencialidade, mais uma vez, testes à API foram realizados. Contudo, os testes executados tiveram critérios diferentes de validação, o que por sua vez implicou analisar respostas diferentes. Neste contexto, é esperado, por exemplo, que seja obtido um código HTTP 401 que reflete um acesso não autorizado, ou um código HTTP 403 que se traduz para acessos proibidos por falta de permissões.

Realizaram-se testes para observar se era possível aceder às funcionalidades do sistema sem efetuar uma autenticação prévia, o que resultou em sucesso visto que o acesso era sempre negado (token de acesso não foi atribuído ou era inválido) - Código 401 (Não Autorizado).

Para testar as permissões de acesso após a autenticação ser feita, realizaram-se pedidos às funcionalidades das quais apenas gestores ou administradores do FireLoc eram autorizados, os pedidos foram feitos em condições que não fosse possível o acesso.

Para tal, efetuou-se o processo de autenticação com um utilizador registado sem permissões associadas (utilizador comum) e procedeu-se a execução de pedidos às funcionalidades cujas permissões de gestor ou administrador eram exigidas. Como era esperado, o resultado dos testes também foram satisfatórios. As respostas obtidas indicavam sempre que as ações não se poderiam executar por falta de permissões - Código 403 (Acesso Proibido).

Desta forma, pode-se assumir que a confidencialidade no sistema FireLoc é garantida, uma vez que os acessos são todos negados caso as condições de acesso não sejam devidamente cumpridas (autenticação feita com sucesso ou permissões associadas).

6.3.2 Testes da Disponibilidade no Sistema

A disponibilidade é, fortemente, influenciada pelo hardware que o sistema possui. Apesar desta dependência de tecnologia física, o sistema FireLoc está a utilizar, como já foi referido, serviços da AWS, o que dá garantias de que o hardware irá comportar-se satisfatoriamente. Para medir a disponibilidade no sistema, realizou-se a monitorização da API, de forma que se pudessem detetar intervalos de tempo em que o sistema não se encontre acessível.

A monitorização foi executada pelo Postman, uma vez que possui ferramentas para observar continuamente a “vitalidade” da API. Esta ferramenta notifica os resultados semanalmente indicando as falhas ou períodos em que o sistema não respondeu aos pedidos.

A monitorização foi realizada por um período, de apenas, um mês, com início no dia 9 de Maio de 2022 às 21h00 e finalizado no dia 9 de Junho de 2022 às 21h00. Idealmente, os testes de disponibilidade devem ser feitos pelo período mínimo de um ano, de modo que os resultados sejam mais precisos com a realidade. Apesar do período de tempo ter sido curto, os resultados obtidos através da monitorização foram bastante satisfatórios uma vez que foi registada uma disponibilidade de 99.18%. O motivo por não se ter registado uma disponibilidade de 100% foi pelo facto dos servidores terem sido reiniciados após atualizações do sistema (manutenção), estando em downtime por, aproximadamente, 5 horas.

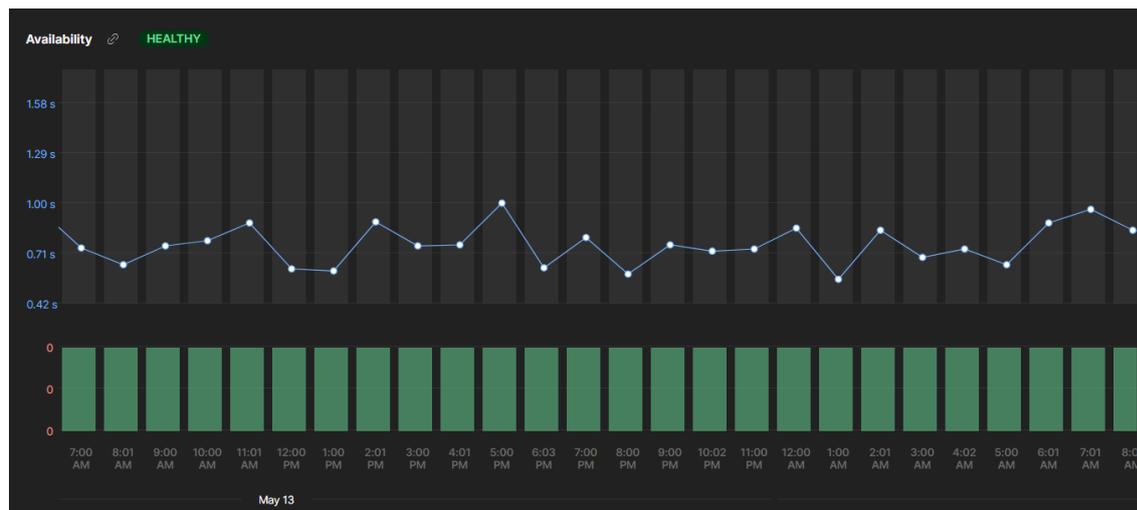


Figura 6.4: Medição da Disponibilidade - Monitorização da API FireLoc

Para fins de representação, torna-se difícil representar o comportamento da API em termos de disponibilidade durante 31 dias. Apesar disso, é possível observar na figura 6.4 que o sistema encontra-se “healthy” e sem erros encontrados durante todo o período de avaliação. Desta forma, pode-se assumir que a disponibilidade no sistema FireLoc é garantida, uma

vez que cumpriu com o valor definido de 99% de tempo, no cenário deste atributo de qualidade.

6.3.3 Testes da Interoperabilidade no Sistema

No Capítulo 5, mais especificamente na secção 5.1.12, descreveu-se os componentes desenvolvidos em relação ao GeoServer. Foi definido no cenário da interoperabilidade que a sua medição é realizada através da interação da API FireLoc com o GeoServer. Receber uma resposta do gerador de serviços com o conteúdo desejado certifica que o sistema é interoperável. A integração de protocolos da OGC tem sido um critério muito importante na escolha das tecnologias para que, efetivamente, não houvessem inconvenientes no uso de tecnologias GIS.

Desta forma, os componentes desenvolvidos em relação ao GeoServer, mais precisamente a obtenção de serviços, foram usados para testar a interoperabilidade do sistema. A realização dos testes tiveram em conta dois protocolos da OGC: WMS e WFS. Ambos os protocolos retornam referências de dados geospaciais, e que por sua vez são utilizados no lado cliente (Leaflet) para renderizar esta informação nos mapas. Mais detalhes sobre os testes de interoperabilidade são abordados na secção seguinte.

6.3.4 Testes de Performance no Sistema

A medição de performance no sistema FireLoc foi medida usando duas técnicas diferentes, uma vez que se estão a considerar dois tipos de performance: Renderização de informação geospacial, e Tempo de resposta dos pedidos à API.

Performance - Tempo de resposta da API

Em relação ao tempo de resposta da API, este já se encontra garantido implicitamente através dos resultados obtidos na secção 6.1.5. A figura 6.3 representa os resultados obtidos, na qual é evidente que o tempo médio de resposta da API está dentro dos valores definidos no cenário do atributo (inferior a 1 segundo). Desta forma, pode-se assumir que a performance relativamente ao tempo de resposta da API no sistema FireLoc é garantida, uma vez que cumpre com o tempo médio de resposta é inferior a 1 segundo.

Performance - Renderização de dados geospaciais

Para evitar problemas de performance na renderização de dados geospaciais nos mapas, devido a sua grande complexidade e dimensão, podem ser aplicadas uma série de estratégias no lado cliente e servidor.

Tile Caching

Uma das técnicas do GeoServer é a estratégia de Tile Caching via GeoWebCache. Tile Caching pode ser definido como pequenas porções de um serviço gerado pelo GeoServer que permitem às plataformas de renderização de mapas, como o Leaflet, renderizar a informação do serviço mais rapidamente no mapa, sem ter de realizar o processamento no

servidor em tempo real. Tile Caching é bastante complexo e, por esse motivo, irão ser revelados novos conceitos relacionados com esta técnica de renderização.

Tile Caching é baseado em *render schemes* por grelhas de referência. As células que compõem a grelha podem ser ajustadas através de um aumento ou diminuição da ampliação de visualização, conceitos conhecidos popularmente como *zoom in* e *zoom out*, respetivamente. Esta técnica pode ser implementada pelo GeoWebCache, que é uma aplicação independente integrada no GeoServer que permite que os serviços REST do mesmo, sejam adaptados para segurar Tile Caching nas suas respostas. Para o processo ficar mais claro é representado na figura 6.5 o esquema aplicado por esta técnica de renderização.

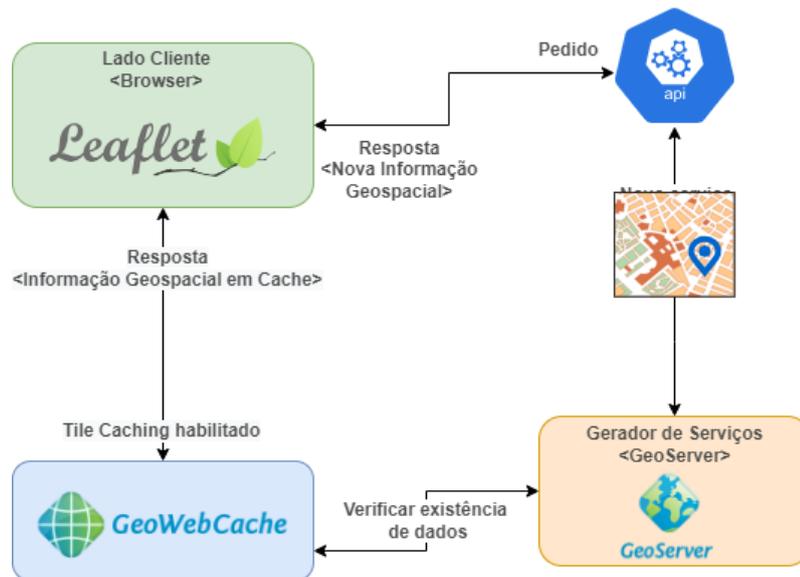


Figura 6.5: Esquema de processamento com GeoWebCache

No esquema apresentado pode-se observar que, quando o lado de cliente efetua um pedido, o GeoServer verifica se existe informação na Cache para uma Tile em condições específicas de ampliação, ou seja, é verificado se existe uma representação para o nível de *zoom* especificado na cache. Quando não se verifica a existência dessas condições, o GeoServer gera uma nova Tile para o nível de *zoom* solicitado e é devolvido para o cliente. Este novo Tile é armazenado em Cache para que seja automaticamente renderizado se o pedido voltar a ser feito. Este processo permite descartar pedidos que procurem gerar serviços desnecessários e, desta forma, poupar recursos, tempo de processamento no servidor e aumentar, consequentemente, a performance de renderização dos dados.

Apenas realizou-se testes sem a aplicação do GeoWebCache por questões de prioridade de desenvolvimento (desenvolvimento de módulos da API). Contudo, proporcionou-se um contributo importante para avaliação de performance de renderização e observações para a melhorar. Os testes foram feitos na aplicação web criada, mencionada na secção 5.2.1, para observar o comportamento dos serviços gerados no GeoServer.

Leaflet

Nesses testes, verificou-se, sem nenhuma surpresa, que o número de camadas renderizadas influencia a performance de navegação no mapa do Leaflet. A partir das 5000 camadas, o Leaflet começa a apresentar pequenos atrasos de resposta o que prejudica, consideravelmente, as tarefas dos utilizadores. Este atraso deve-se ao facto de cada uma das camadas

representar um objeto do Document Object Model (DOM). Isto significa que, se tivermos 5000 camadas, teremos 5000 objetos no DOM, o que naturalmente prejudica a navegação no browser. Para resolver este problema, foi usado um *plug-in* externo do Leaflet, o Marker Clustering. Este *plug-in* aplica uma técnica de agrupamento de camadas que estão próximas umas das outras num determinado nível de *zoom*.



Figura 6.6: Marker Clustering - Exemplo

Como se pode observar na figura 6.6 estão renderizadas 5000 camadas agrupadas em 13 *clusters* facilitando, de forma bastante perceptível, a renderização. Os tempos desceram de 8080 ms para 210 ms, equivalente a uma descida brutal de 3847%.

Serviços com Filtragem

Apesar do *clustering* ser uma boa solução, o sistema FireLoc pode melhorar, ainda mais, a sua performance de renderização. Foram feitas algumas experiências com a criação de serviços WFS do GeoServer com parâmetros de filtragem, nomeadamente, com filtros Common Query Language (CQL). Os parâmetros em estudo foram a área de visualização e janela temporal. Este estudo deve-se a questões de simulação sobre situações extremas em que poderiam chegar a existir milhares de contribuições e ocorrer uma sobrecarga no sistema.

Filtragem com BBOX

A filtragem com BBOX (Bounding Box) pode ser definida como uma caixa delimitadora. Neste contexto, essa caixa é a área de visualização no mapa, e é criada a partir de duas coordenadas: o vértice inferior esquerdo, e o vértice superior direito. Este tipo de filtragem permite que apenas sejam acedidas as geometrias que se encontrem dentro do BBOX, sendo que as restantes serão descartadas. Para efeitos de aplicabilidade da técnica, os pedidos ao GeoServer têm o filtro CQL quando o utilizador faça zoom in/out e navegue pelo mapa. Isto significa que os pedidos são gerados dinamicamente com a navegação.

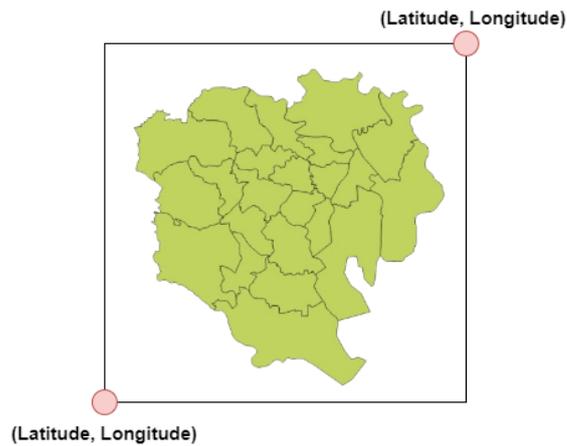


Figura 6.7: Bounding Box - Figura adaptada de [45]

Filtragem com Operadores

A filtragem com operadores é mais simples do ponto de vista de implementação e serão realizados sobre os atributos que digam respeito às datas de ocorrência. Desta forma, são filtradas as geometrias que estejam dentro de uma janela temporal (data início e data fim).

Estas técnicas, em termos de performance, melhoraram o tempo de renderização dos dados geospaciais nos mapas consideravelmente, visto que haveriam menos objetos para renderizar. Todos os testes provaram que a interoperabilidade está garantida no sistema. Não houve nenhuma barreira entre a comunicação Leaflet \longleftrightarrow API FireLoc \longleftrightarrow GeoServer.

Capítulo 7

Conclusão

Ao longo do ano letivo 2021/2022, houve contacto direto com o ambiente de trabalho, desenvolvendo atitudes fundamentais e profissionalizantes com duração, supervisão e comunicação constante com colegas experientes, o que possibilitou uma evolução em termos pessoais e profissional. Dito isto, neste último capítulo irá ser feita uma comparação entre o plano de trabalhos idealizado e o plano de trabalhos efetivo, e uma reflexão sobre toda a experiência adquirida que a participação neste projeto possibilitou, bem como observações para trabalho futuro.

7.1 Análise do Planeamento

No início desta dissertação, foi preparado e apresentado o plano de trabalhos a realizar, de forma que ficasse claro e registado as categorias de atividades e o tempo previsto para concluir as tarefas. Contudo, no decorrer da dissertação o plano de trabalhos sofreu alterações em termos de tempo, tarefas e ordem de execução das mesmas.

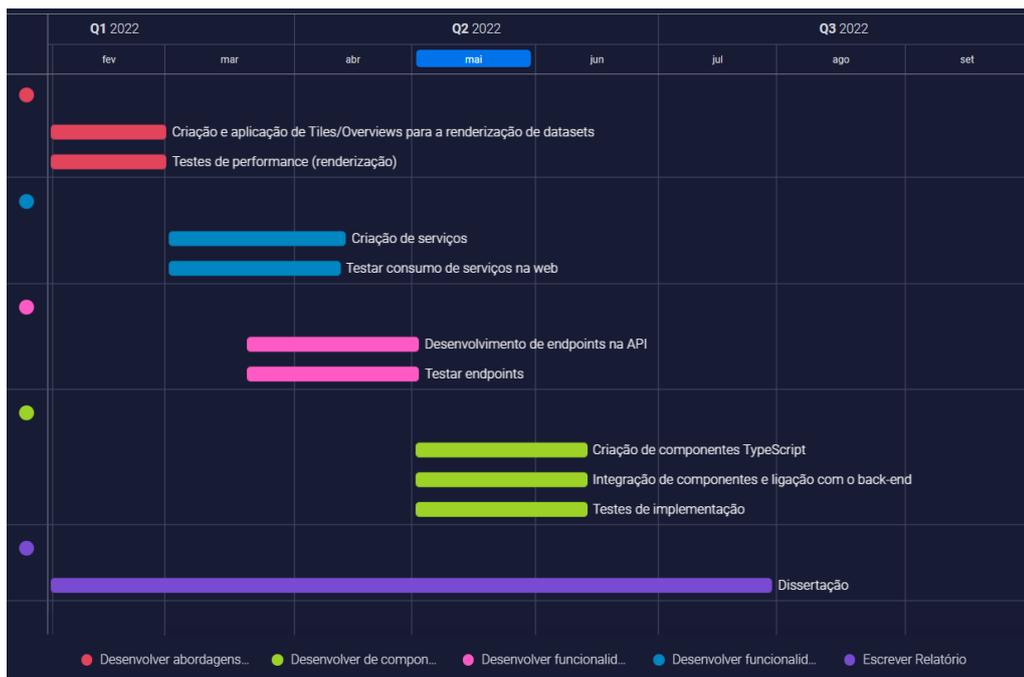


Figura 7.1: Diagrama de Gantt - Plano idealizado

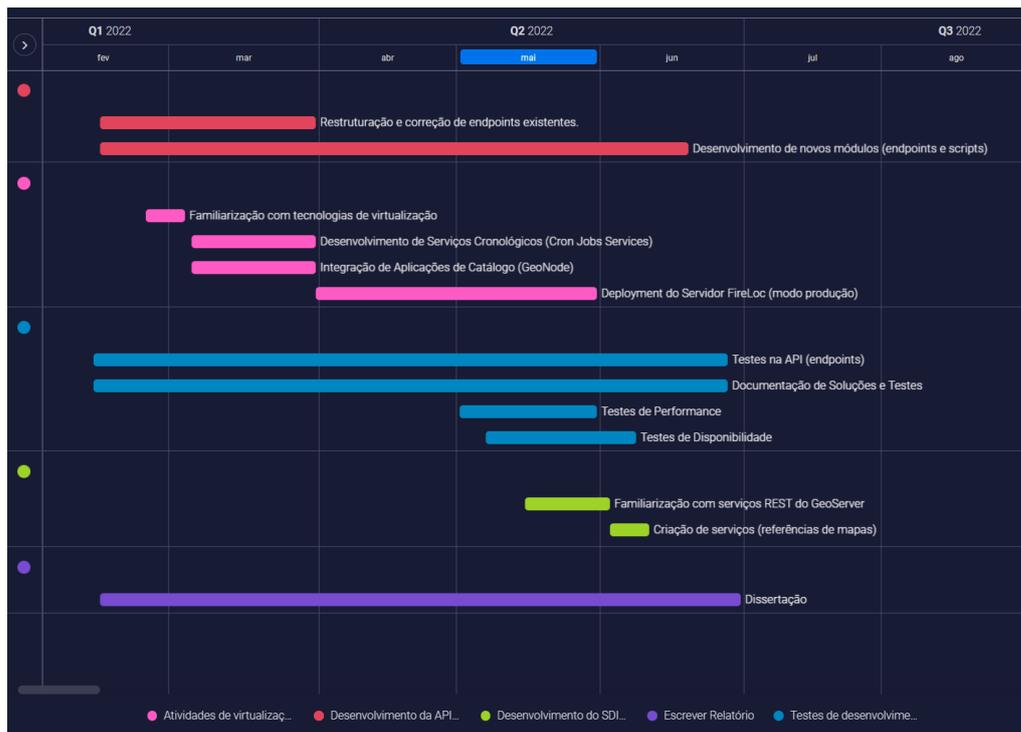


Figura 7.2: Diagrama de Gantt - Plano efetivo

Na figura 7.1 está representado o plano de trabalhos idealizado e na figura 7.2 está representado o plano efetivo de trabalhos realizado. Como se pode observar na figura 7.2, houve alterações evidentes no plano de trabalho em relação ao que era esperado. As principais razões por terem ocorrido estas alterações foi por questões de prioridade em relação às necessidades do projeto FireLoc.

As alterações não prejudicaram o fluxo de trabalho, o que permitiu reduzir a curva de aprendizagem e apuraram a qualidade e produtividade em termos de trabalho, e que fossem realizadas atividades que não se encontravam definidas no plano de trabalho idealizado. Por outro lado, houve atividades em particular que foram dispensadas do plano, nomeadamente, atividades que implicassem o desenvolvimento da UI no Portal. Poderia haver uma eventual participação nas duas vertentes, front-end e back-end tornando-se num trabalho Full-Stack. Contudo, uma equipa especializada ficou responsável pelo desenvolvimento do lado cliente e estas tarefas foram assumidas por outros colegas de trabalho, o que permitiu que houvesse um foco apenas no lado do servidor.

Em relação à calendarização das tarefas, estas foram realizadas, intencionalmente, desta maneira com supervisão do orientador e coordenador, uma vez que permitia adquirir experiência de forma progressiva para que as tarefas mais complexas fossem simplificadas.

7.2 Reflexão Final

Na fase inicial de integração no projeto, houve alguma preocupação na familiarização de algumas tecnologias que não eram conhecidas, nomeadamente, o Django e o GeoServer. Desconhecer as principais tecnologias do projeto obrigou a realizar um esforço extra, para não comprometer as atividades práticas (desenvolvimento e testes). Ambos os orientadores, mostraram ser muito compreensivos e disponibilizaram-se, durante todo o ano letivo, para atender as questões e auxiliar nas dificuldades que foram sendo sentidas. Esta cooperação facilitou muito a integração na equipa, na participação de reuniões e, conseqüentemente, possibilitou um bom ambiente de trabalho.

Entre as maiores dificuldades, pode-se destacar aquelas relacionadas com tecnologias GIS e as tecnologias de virtualização. Apesar da familiaridade com alguns conceitos de tecnologias GIS, a participação no projeto FireLoc possibilitou desenvolver competências neste tipo de tecnologia e tipos de dados de forma profissional e robusta. A manipulação de informação geospacial é complexa, pois exige que seja feito um grande pré-processamento e análise. Muitos conceitos e técnicas são triviais para pessoas especializadas na área da Geografia, outras na área de Informática e conciliar ambas competências foi um grande desafio, tendo sido uma das principais metas pessoais.

Por outro lado, as tecnologias de virtualização são relativamente “recentes” em termos de popularidade. Este tipo de ferramenta tem vindo a ser muito usada nos últimos anos, sendo uma mais valia para as entidades que adotam o seu uso. O processo de containerização de aplicações exige conhecimentos avançados de OS e experiência em linguagens de programação de alto nível como o Bash. Todas estas condições, no seu conjunto, tornaram as atividades com Docker e *deployment* bastante complexas, principalmente pela inexperiência.

7.3 Trabalho Futuro

Como foi dito anteriormente, está a ser desenvolvido o Portal como aplicação web para o sistema FireLoc, assim como uma versão para iOS para a aplicação móvel. Estas aplicações são cruciais para que o sistema seja acessível para um maior conjunto de utilizadores e, por sua vez, ter fontes de dados mais aprimoradas.

As aplicações de catálogo merecem uma dedicação apropriada para evitar desperdiçar recursos e tempo de desenvolvimento na aplicação de back-office. Apesar de terem sido abordadas na dissertação, estas interfaces tiveram uma fração de atividades bastante reduzida e não houve grande realce na sua importância. Estas aplicações são, sem dúvida, algo a ter em conta para trabalho futuro.

Para além do desenvolvimento das aplicações, é importante destacar que a API FireLoc não se encontra desenvolvida na totalidade, pelo que mais módulos deverão ser desenvolvidos e/ou melhorar os existentes, de acordo com as necessidades do sistema. Todos os módulos apresentados, foram documentados e sujeitos a um conjunto de testes para garantir que o seu funcionamento vai de encontro ao esperado. Estas tarefas são importantes e deverá ser dado o seguimento destas práticas no futuro para auxiliar novos membros que integrem a equipa FireLoc e/ou recordar aspetos técnicos aos membros efetivos.

Embora os testes unitários sejam importantes, o passo seguinte será, naturalmente, realizar testes de integração com as aplicações móvel e web de modo a que o sistema se encontre funcional e disponível para a sua utilização. Alguns módulos validados já estão a ser

integrados na aplicação web e é esperado que todos se encontrem integrados até o final do ano de 2022.

Uma vez que toda a vertente funcional seja implementada e testada, faz sentido que a parte qualitativa seja algo a ter em conta no futuro, isto é, garantir mais atributos de qualidade e/ou reforçar os atributos existentes. A confiabilidade é um dos atributos mais importantes em qualquer sistema de software, pois é a garantia de um serviço correto. A confiabilidade não pode ser testada diretamente pois, tal como a Segurança, é um conceito.

Deste modo, este atributo de qualidade é medido através do conjunto de outros atributos como Disponibilidade, Segurança, Integridade e Manutenibilidade (facilidade de manutenção de um sistema). É de notar que muitos atributos que constituem a Confiabilidade foram apresentados e testados, logo faz sentido que os restantes atributos sejam testados de modo a poder garantir um serviço confiável. Outro atributo importante é a Performance em termos de Renderização. A aplicação de estratégias como Cluster Rendering e Filtragem são, apenas, algumas técnicas. Deve ter-se em consideração outras, como Tile Caching, de forma que o sistema esteja preparado para qualquer situação que envolva um sobrecarregamento do sistema e os seus serviços falhem.

Para terminar, é importante referir que todo o desenvolvimento da API tem em conta a reutilização de métodos em termos funcionais, isto é, o Firegis. O uso de tecnologias como Angular, baseados em componentes, indica, também, que haverá um regime de reutilização dos mesmos. A ideia é tornar o sistema completamente modular, dividido em componentes reutilizáveis e dinâmicos, para que, no futuro, o desenvolvimento e manutenção seja mais claro e simplificado.

Referências

- [1] Github - leaflet/leaflet. <https://github.com/Leaflet/Leaflet>.
- [2] PostgreSQL 14. Appendix m. glossary. In *Consistency*, <https://www.postgresql.org/docs/current/glossary.html#GLOSSARY-CONSISTENCY>, 2022.
- [3] Domingos Xavier Viegas Abdelrahman Abouali. Fire ros calculator. In *A Tool to Measure the Rate of Spread of a Propagating Wildfire in a Laboratory Setting*, <https://openresearchsoftware.metajnl.com/articles/10.5334/jors.221/>, 2019.
- [4] Vladimir Agafonkin. Leaflet. <https://leafletjs.com/>, 2021.
- [5] Aqua. Cloud native. In *Docker Containers vs. Virtual Machines*, <https://www.aquasec.com/cloud-native-academy/docker-container/docker-containers-vs-virtual-machines/>, 2022.
- [6] ArcGIS. Digital elevation models. <https://learn.arcgis.com/en/related-concepts/digital-elevation-models.htm>, 2021.
- [7] ArcGIS. Coordinate systems, projections, and transformations. <https://pro.arcgis.com/en/pro-app/2.7/help/mapping/properties/coordinate-systems-and-projections.htm>, 2022.
- [8] bluehost. Cron job basics. In *What is a Cron Job?*, <https://www.bluehost.com/help/article/cron-job-basics>, 2019.
- [9] Gonçalo Castelo Branco. Logística florestal. In *Wayback Machine*, https://web.archive.org/web/20070806120706/http://www.logistica-florestal.pt/maillinglist/newsletters/20050705_nlCompleta9_pt.html, 2005.
- [10] Clear Bridge. 5 key benefits of native mobile app development. <https://clearbridgemobile.com/benefits-of-native-mobile-app-development/>, 2020.
- [11] Brandon Chen. What you need to know. In *PostgreSQL vs. MySQL*, <https://fivetrn.com/blog/postgresql-vs-mysql>, 2021.
- [12] INESC Coimbra. Projeto fireloc onde esta o fogo In *Faculdade de Ciências e Tecnologia*, <https://fireloc.org/>, 2019.
- [13] Open Geospatial Consortium. Web feature service. <https://www.ogc.org/standards/wfs>, 2021.
- [14] Open Geospatial Consortium. Web map service. <https://www.ogc.org/standards/wms>, 2021.

- [15] Open Geospatial Consortium. Your global resource for geospatial information and standards. In *The Home of Location Technology Innovation and Collaboration*, <https://www.ogc.org/>, 2021.
- [16] Open Geospatial Consortium. Open geospatial consortium. In *OGC*, <https://www.ogc.org/>, 2022.
- [17] Instituto da Conservação da Natureza e das Florestas. Territórios arditos. <https://sig.icnf.pt/portal/home/item.htmlid983c4e6c4d5b4666b258a3ad5f3ea5af>, 2022.
- [18] Diário de notícias. Fogo posto e uso negligente são as principais causas dos incêndios deste ano. <https://www.dn.pt/sociedade/fogo-posto-e-uso-negligente-sao-as-principais-causas-dos-incendios-deste-ano-14283586.html>, 2021.
- [19] Governo de Portugal. Conjunto de dados. <https://dados.gov.pt/pt/datasets/>, 2022.
- [20] Darshit Shah Deepika Shukla, Chirag Shivnani. In *Comparing Oracle Spatial and Postgres PostGIS*, <http://csjournals.com/IJCSC/PDF7-2/16.%20Deepika.pdf>, 2016.
- [21] dnoticias. Madeira participa em reunião de arranque do projecto rede gesfogo. <https://www.dnoticias.pt/madeira/madeira-participa-em-reuniao-de-arranque-do-projecto-rede-gesfogo-XG5945990>, 2020.
- [22] Django Documentation. Password management in django. <https://docs.djangoproject.com/en/4.0/topics/auth/passwords/>, 2022.
- [23] esparkinfo. qual você deve escolher em 2021 In *Django vs Laravel vs Node.js*, <https://www.esparkinfo.com/laravel-vs-django-vs-nodejs.html>, 2021.
- [24] Fireloc. Fireloc - localize o fogo. In *Sistema Fireloc*, <https://fireloc.org/>, 2019.
- [25] Jody Garnett. Geonetwork basics. In *FOSDEM*, <https://www.youtube.com/watchv=BHKWlIxaXeI>, 2020.
- [26] GeoNetwork. Geonetwork opensource. <https://geonetwork-opensource.org/>, 2022.
- [27] GeoNode. Learn more about the features of geonode. In *For Administrators*, https://geonode.org/admin_features/, 2021.
- [28] GeoNode. Learn more about the features of geonode. In *For Users*, https://geonode.org/user_features/, 2021.
- [29] GeoNode. Geonode. <https://geonode.org/>, 2022.
- [30] GeoNode. Geonode basic installation. In *GeoNode 3.2.1 documentation*, <https://docs.geonode.org/en/master/install/basic/index.html>, 2022.
- [31] OGC GeoPackage. Geopackage. In *FAQ*, <https://www.geopackage.org/>, 2021.
- [32] GeoServer. Mysql. In *GeoServer 2.21.x User Manual*, <https://docs.geoserver.org/latest/en/user/data/database/mysql.html>, 2021.
- [33] GeoServer. Geoserver stores. In *GeoServer 2.21.x User Manual*, <https://docs.geoserver.org/stable/en/user/data/webadmin/stores.html>, 2022.
- [34] GeoServer. Geoserver workspace. In *GeoServer API Docs*, <https://docs.geoserver.org/latest/en/api/1.0.0/workspaces.yaml>, 2022.

-
- [35] GeoServer. geoserver/geoserver. <https://github.com/MapServer/MapServer>, 2022.
- [36] GesFoGO. Objetivos. In *Geral e Específico*, <https://www.gesfogo.ulpgc.es/index.php/pt/gesfogo-pt/objetivos-pt>, 2020.
- [37] Google. googlemaps/google-maps-services-python. <https://github.com/googlemaps/google-maps-services-python>, 2022.
- [38] Florent Gravin. The 3 best open source web mapping libraries. In *Camptocamp Geospatial Solutions*, <https://dev.to/camptocamp-geo/the-3-best-open-source-web-mapping-libraries-57o7>, 2020.
- [39] Raluca Budiu h. Native apps, web apps, and hybrid apps. In *Mobile*, <https://www.nngroup.com/articles/mobile-native-apps/>, 2013.
- [40] Steve Henty. Ui response times. In *Medium*, <https://medium.com/slhenty/ui-response-times-acec744f3157>, 2015.
- [41] María Jesús Hernández. La ulpgc diseña estaciones móviles de observación para la gestión de incendios. In *Científicos del Idetic participan en la creación de una red tecnológica para el control y el seguimiento del fuego en las áreas forestales de Canarias, Madeira y Cabo Verde*, <https://www.laprovincia.es/sociedad/2020/03/05/ulpgc-disena-estaciones-moviles-observacion-8293026.html>, 2020.
- [42] Hivelocity. What is cron job? <https://www.hivelocity.net/kb/what-is-cron-job/>, 2022.
- [43] Tom Hollands. Geonetwork and spatial metadata cataloguing. In *Spatial Vision*, <https://spatialvision.com.au/geonetwork-and-spatial-metadata-cataloguing/>, 2020.
- [44] IPMA. Interface de programação de aplicações do ipma. In *serviço online*, <https://api.ipma.pt/>, 2022.
- [45] JBOY24. Tóquio mapa imagem vetorial de © jboy24. In *DepositPhotos*, <https://br.depositphotos.com/74647989/stock-illustration-tokyo-map.html>, 2015.
- [46] Gary Scudder e Luk N. Van Wassenhove Joseph Blackburn. Improving speed and productivity of software development. https://flora.insead.edu/fichiersti_wp/inseadwp1996/96-11.pdf, 1998.
- [47] Francisco Rego João Silva e Tiago da Silva João Pereira, José Pereira. Incêndios florestais em portugal. In *Caracterização, Impactes e Prevenção*, https://academia.edu/17357376/Incêndios_florestais_em_Portugal_caracterização_impactes_e_prevenção, 2006.
- [48] Joy Beatty Karl Wieggers. Exploring quality attribute requirements. In *Quality Attribute in Brosseau's spreadsheet*, <https://www.batimes.com/articles/exploring-quality-attribute-requirements/>, 2014.
- [49] MapBox. Bee2fire. In *Sistema de detecção precoce de incêndios e apoio à decisão*, <http://www.bee2firedetection.com/>, 2021.
- [50] MapBox. Mapbox. In *Maps and location for developers*, <https://www.mapbox.com/>, 2021.
- [51] MapBox. mapbox/mapbox-gl-js. <https://github.com/mapbox/mapbox-gl-js>, 2022.
- [52] MapServer. Mapserver. In *Mapfile*, <https://www.mapserver.org/mapfile/>, 2021.

- [53] MapServer. Mapserver/mapserver. <https://github.com/MapServer/MapServer>, 2022.
- [54] MetaCarta. Openlayers. <https://openlayers.org/>, 2021.
- [55] Cairo Noletto. Banco de dados: Tipos o que é e suas diferenças! In *Os 6 principais bancos de dados mais usados!*, <https://blog.betrybe.com/tecnologia/bancos-de-dados/>, 2020.
- [56] Fernando Miguel Granja Martins C.M.G. Pedras Rui Lança Elisa Silva Lorenna M Zavala Antonio Jordán Cristina Soares Nuno André Ramos Simões, Helena Paixão. Webgis platform addressed to forest fire management methodologies. https://researchgate.net/publication/312027553_WebGIS_Platform_Addressed_to_Forest_Fire_Management_Methodologies, 2015.
- [57] OpenLayers. openlayers/openlayers. <https://github.com/openlayers/openlayers>, 2022.
- [58] OSGeo. Geos. In *Geometry Engine, Open Source*, <https://trac.osgeo.org/geos>, 2021.
- [59] Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones. Parceiros. In *GesFoGO*, <https://www.gesfogo.ulpgc.es/index.php/pt/socios-pt>, 2020.
- [60] Associação para o Desenvolvimento da Aerodinâmica Industrial. Centro de estudos sobre incêndios florestais. In *Centros de Investigação*, <https://www.adai.pt/site/?module=publicRdtarget=centerDetailsid=2centerId=1>.
- [61] Associação para o Desenvolvimento da Aerodinâmica Industrial. Imfire. In *Introdução*, <https://adai.pt/imfire/pt/entrada/>, 2020.
- [62] João Pina. Fogos.pt. In *Ocorrências de incêndio em Portugal*, <https://fogos.pt/sobre>, 2015.
- [63] Google Maps Platform. Pricing plans and api costs. In *Pricing*, <https://mapsplatform.google.com/pricing/>, 2021.
- [64] PORDATA. Estatísticas sobre portugal e europa. In *O que são NUTS?*, <https://www.pordata.pt/O+que+sao+NUTS>, 2022.
- [65] VOST Portugal. Vost portugal. In *Voluntários digitais em situações de emergência*, <https://vost.pt/sobre-a-vost-portugal/>, 2019.
- [66] Fernando Quadro. Introdução ao geoserver. In *GeoServer - Blog do Fernando Quadro*, <https://fernandoquadro.com.br/html/geoserver/>, 2019.
- [67] Raysa Roque and Amber Thomas. Geospatial inconsistency issues in geothermal analysis. In *Geographic and Projected Coordinate Reference Systems*, <https://publications.mygeoenergy.org/grc/1034055.pdf>, 2018.
- [68] ServiceTonic. 10 benefits of using a service catalog. <https://www.servicetonic.com/service-desk/10-benefits-of-using-a-service-catalog-in-it-management/>, 2021.
- [69] Golden Software. Explanation of osm, wms, wfs and wcs servers. <https://support.goldensoftware.com/hc/en-us/articles/115003042608-Explanation-of-OSM-WMS-WFS-and-WCS-servers>, 2021.
- [70] SpatialPost. Mapserver vs geoserver. In *Which One is Better for You*, <https://www.spatialpost.com/mapserver-vs-geoserver/>, 2021.

- [71] stackshare. Mapbox. In *Reviews, Pros Cons*, <https://stackshare.io/mapbox>, 2019.
- [72] Web2App Info Tech. Comparison: Native vs hybrid vs web. <https://www.web2appinfotech.com/mobile-app-comparison-native-vs-hybrid-vs-web/>, 2020.
- [73] P. J. Hyett Scott Chacon Tom Preston-Werner, Chris Wanstrath. Github. <https://github.com/>, 2022.
- [74] OS TÉCNICOS. Nuts. <http://tecnicosessps.blogspot.com/p/nuts.html>.
- [75] Arditi ULPGC. Atividades. In *GesFoGO*, <https://www.gesfogo.ulpgc.es/index.php/pt/atividades-pt>, 2021.
- [76] Carlos Viegas. 1 st year results and milestones. In *IMfire*, <https://adai.pt/imfire/wp-content/uploads/2021/05/20210507-2%C2%BAWS-Projetos-FCT.pdf>, 2021.
- [77] OpenStreetMap wiki. Rendering. <https://wiki.openstreetmap.org/wiki/Rendering>, 2021.
- [78] OpenStreetMap wiki. Openstreetmap. <https://www.openstreetmap.org/about>, 2022.
- [79] Wikipédia. Geographic resources analysis support system. In *GRASS*, <https://pt.Wikipédia.org/wiki/GRASS>, 2020.
- [80] Wikipédia. Geomática. <https://pt.wikipedia.org/wiki/Geomática>, 2020.
- [81] Wikipédia. Geopackage. In *Format*, <https://en.Wikipédia.org/wiki/GeoPackage>, 2021.
- [82] Wikipédia. Spatial data infrastructure. https://en.wikipedia.org/wiki/Spatial_data_infrastructure, 2021.
- [83] Wikipédia. Spatial reference system. In *CRS*, https://en.Wikipédia.org/wiki/Spatial_reference_system, 2021.
- [84] Wikipédia. Web coverage service. https://pt.wikipedia.org/wiki/Web_Coverage_Service, 2021.
- [85] Wikipédia. Web mapping. https://en.Wikipédia.org/wiki/Web_mapping, 2021.
- [86] Wikipédia. Object-relational mapping. In *ORM*, https://en.wikipedia.org/wiki/Object-relational_mapping, 2022.
- [87] Wikipédia. Well-known text representation of geometry. In *Geometry primitives (2D)*, https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry, 2022.
- [88] Jean-Pierre Ybert ÁRita Scheel, Maria Dulce Gaspar. Antracologia, uma nova fonte de informações para a arqueologia brasileira. <https://www.revistas.usp.br/revmae/article/download/109250/107721>, 2016.
- [89] Áurea Marília Madureira e Carvalho. Palinologia forense. https://apcforenses.org/page_id502, 2022.

Apêndices

Esta página foi deixada em branco intencionalmente..

Apêndice A - Diagramas de UC

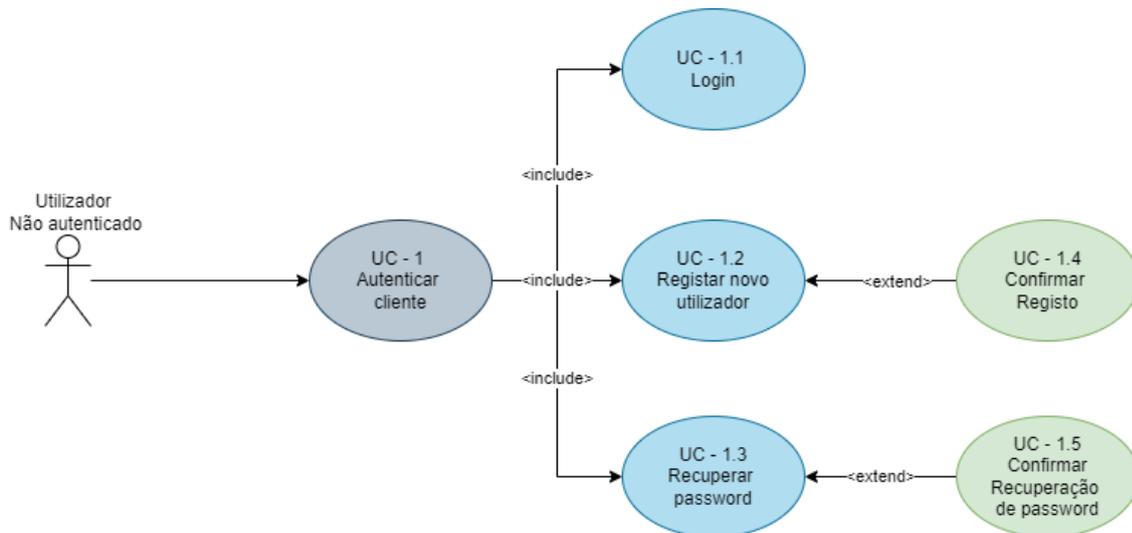


Figura 3: Diagrama UC - 1: Autenticação

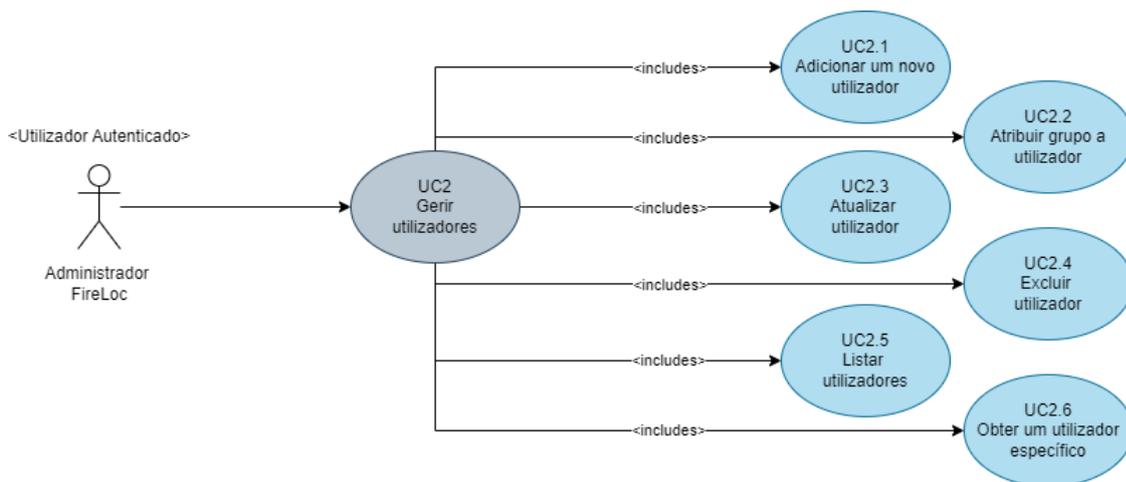


Figura 4: Diagrama UC - 2: Gerir utilizadores

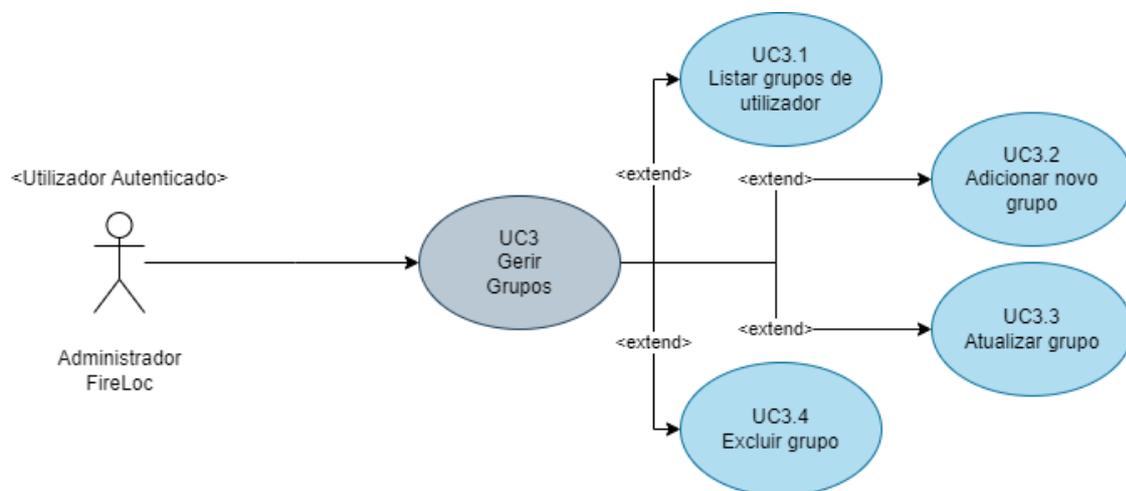


Figura 5: Diagrama UC - 3: Gerir grupos

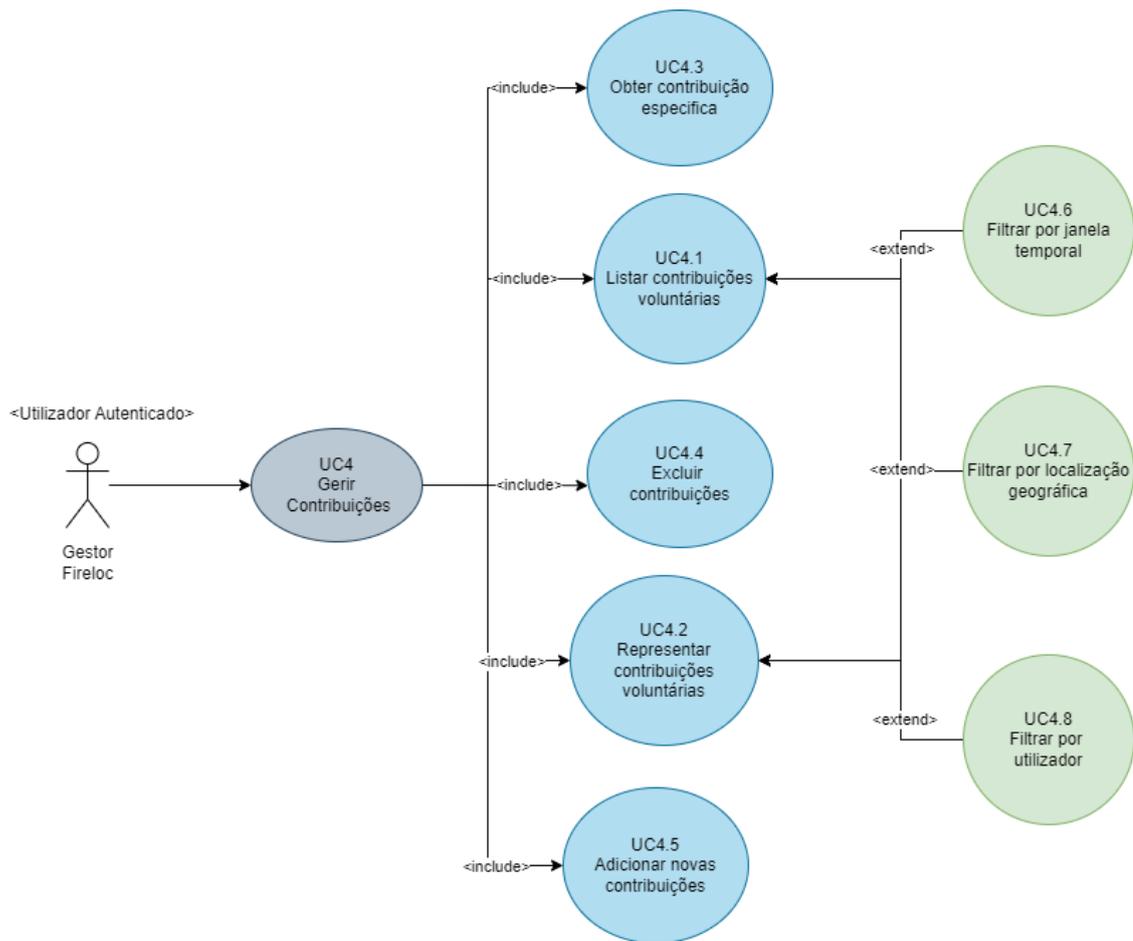


Figura 6: Diagrama UC - 4: Gerir Contribuições

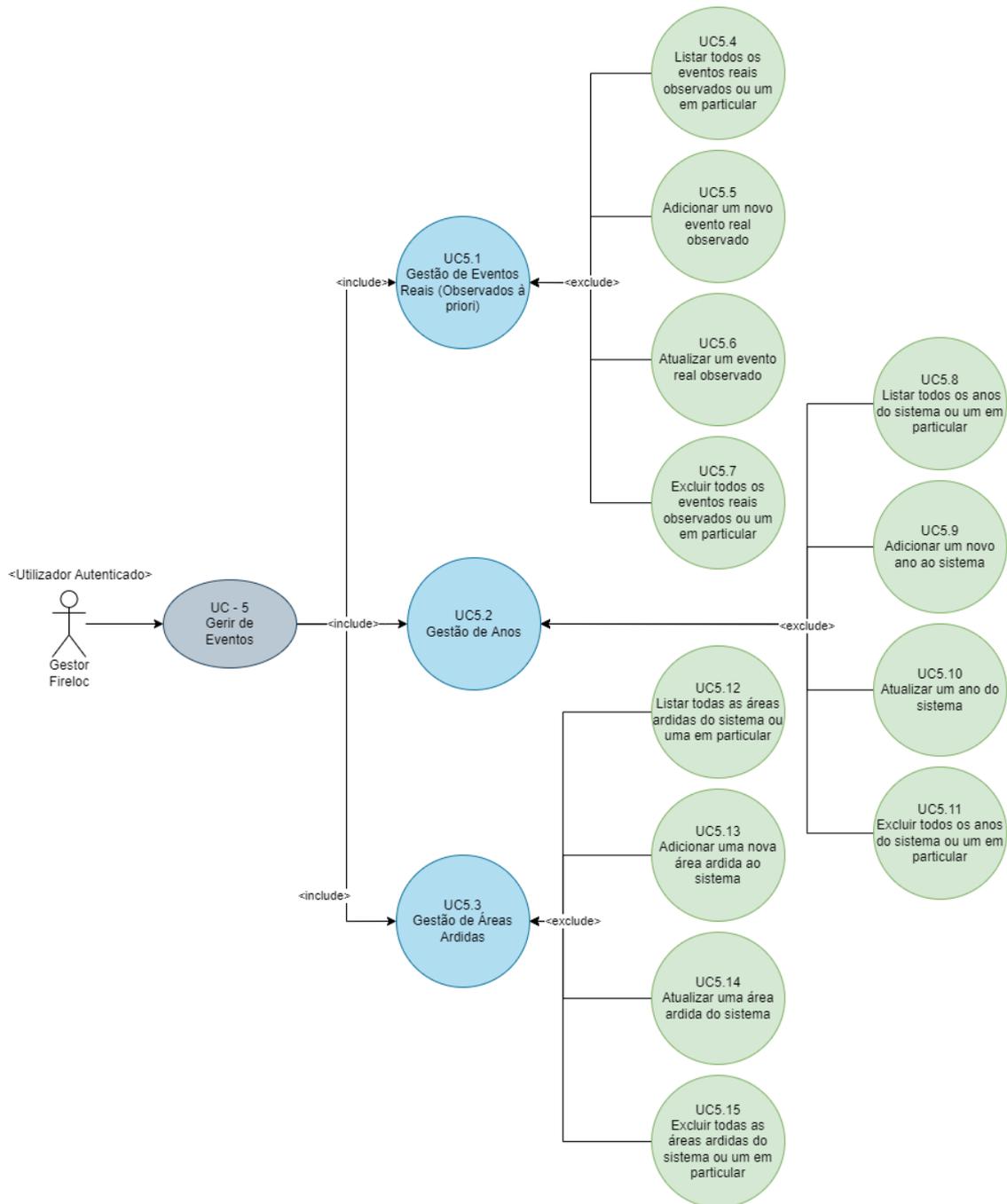


Figura 7: Diagrama UC - 5: Gerir Eventos

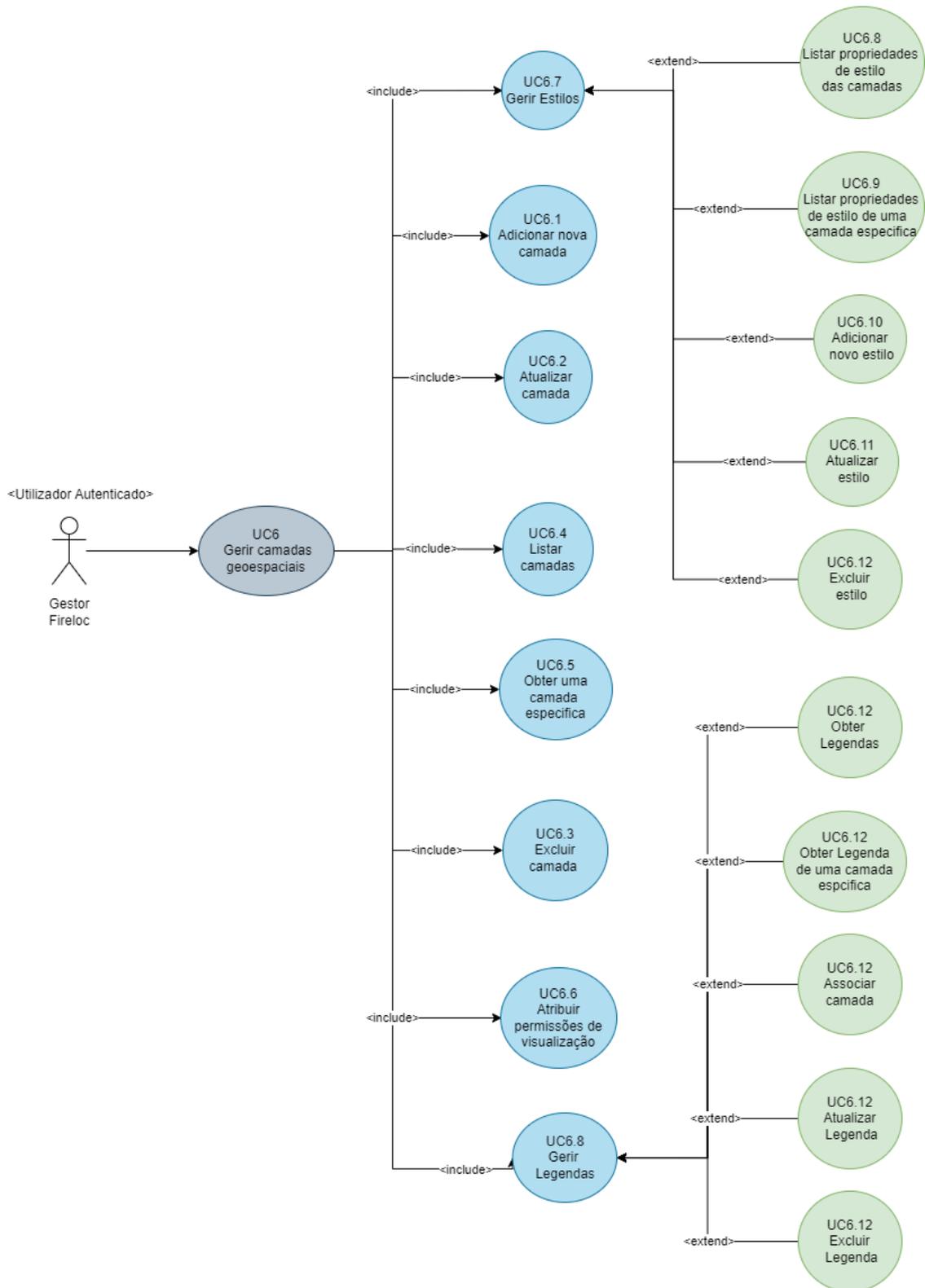


Figura 8: Diagrama UC - 6: Gerir Camadas Geoespaciais

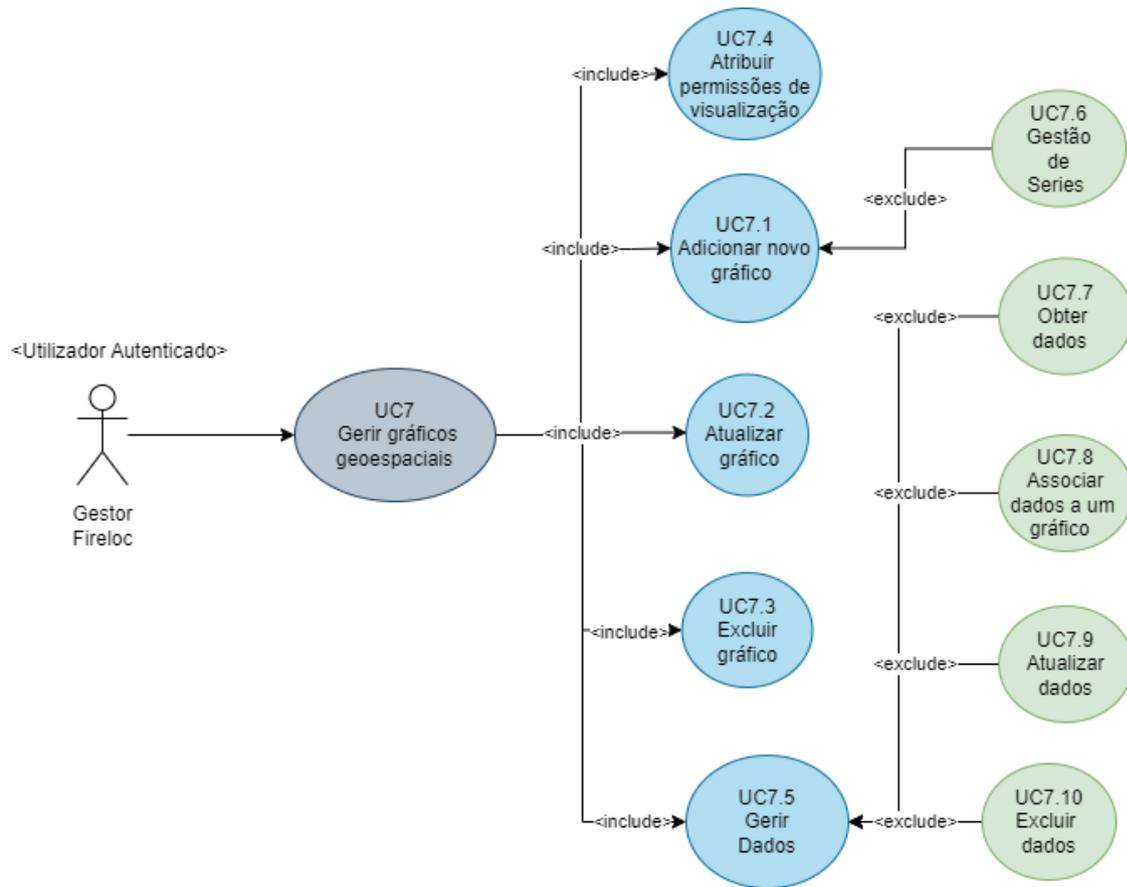


Figura 9: Diagrama UC - 7: Gerir gráficos

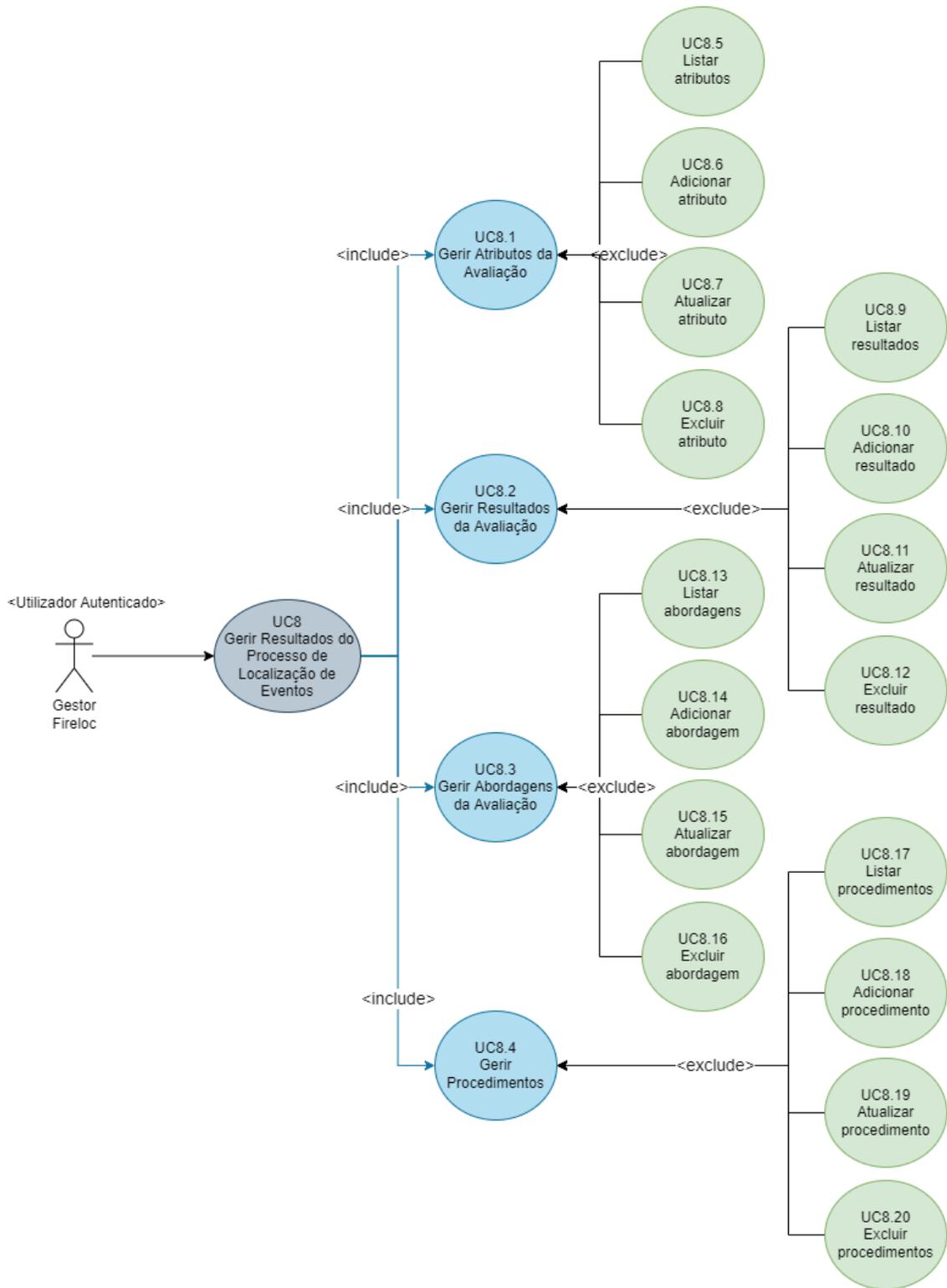


Figura 10: Diagrama UC - 8: Gerir Resultados do Processo de Localização de Eventos

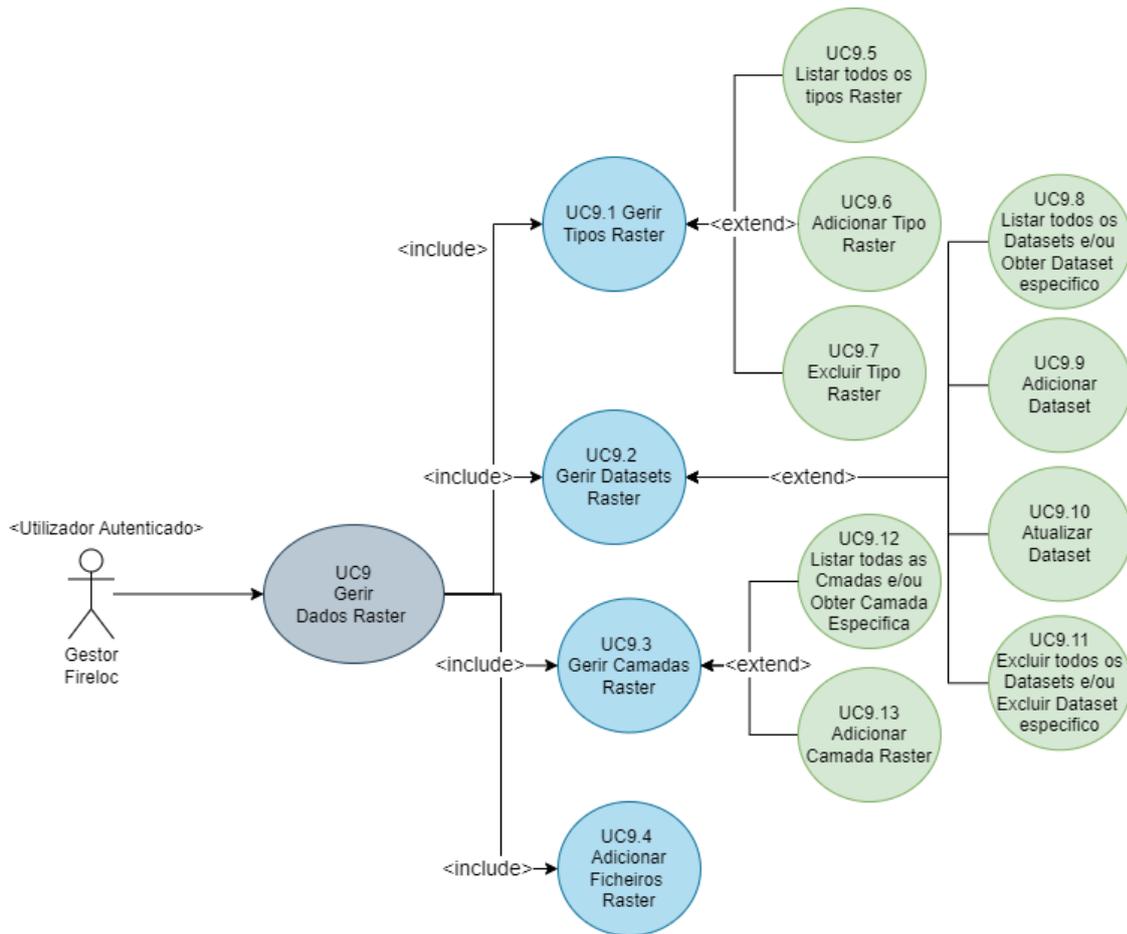


Figura 11: Diagrama UC - 9: Gerir Dados Raster

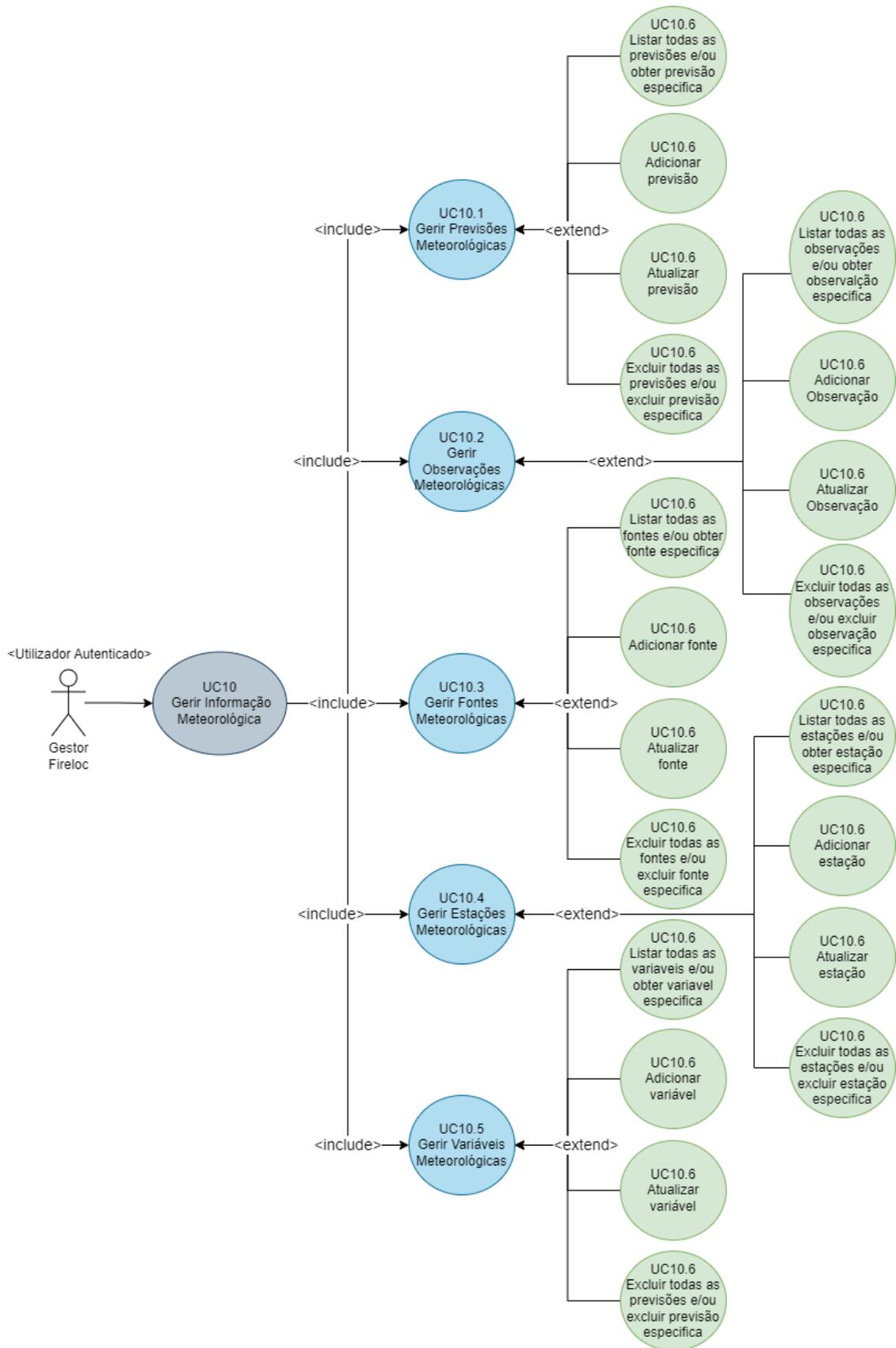


Figura 12: Diagrama UC - 10: Gerir Gerir Informação Meteorológica

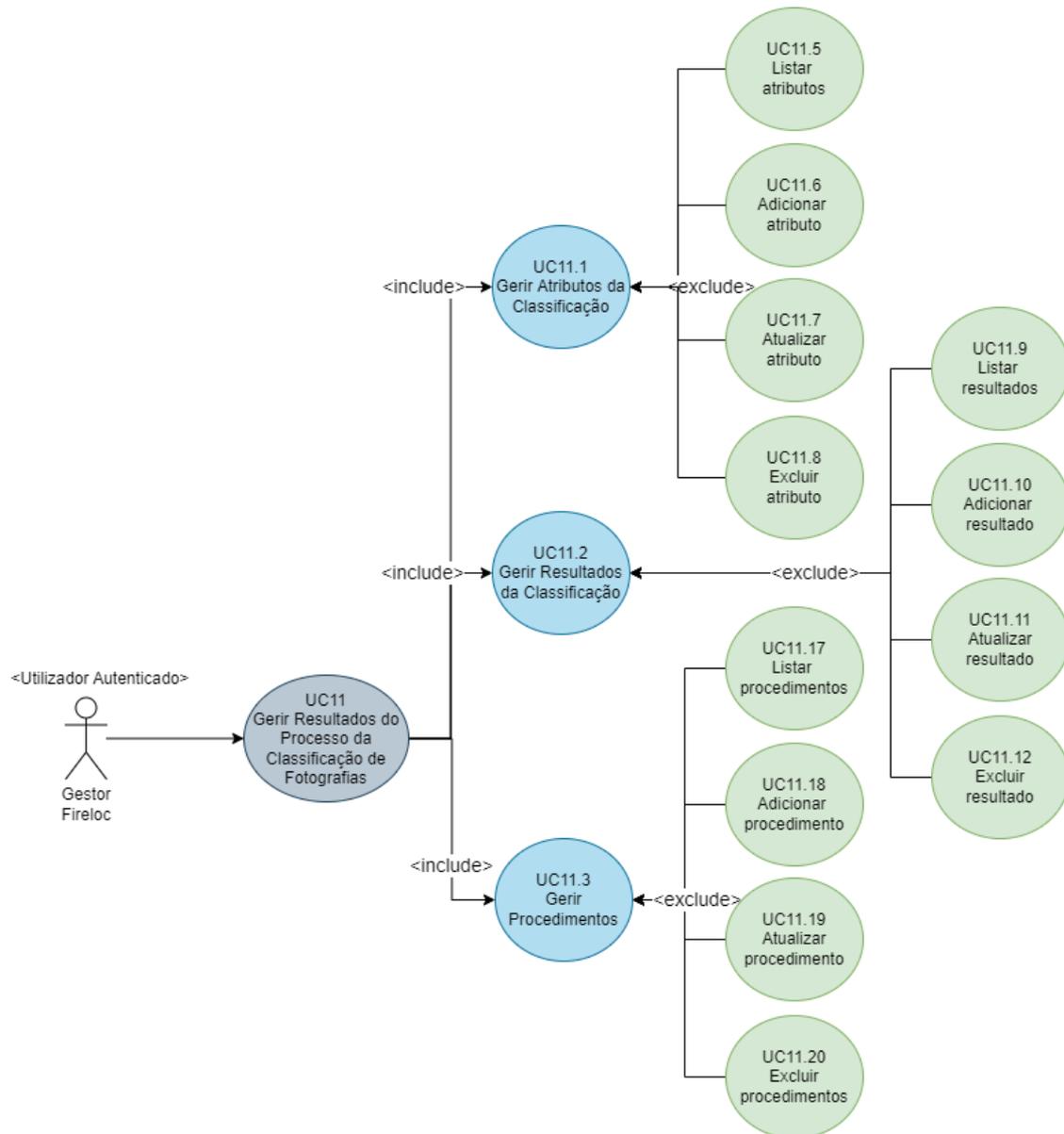


Figura 13: Diagrama UC - 11: Gerir Resultados do Processo da Classificação de Fotografias

Apêndice B - Use Cases

UC 1.1 - Login

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Utilizador não autenticado

Trigger: O utilizador não autenticado deseja aceder ao sistema.

Pré-condições:

1. Ter acesso a internet.

Garantia mínima: O utilizador não autenticado não consegue aceder ao sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O utilizador não autenticado consegue autenticar-se com sucesso e aceder ao sistema.

Cenário principal de sucesso:

1. O utilizador insere as suas credenciais.

2. O sistema valida as credenciais do utilizador.

3. O utilizador é redirecionado para a página principal da aplicação.

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.

Tabela 1: Use Case 1.1 - Login

UC 1.2 - Registrar

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Utilizador não autenticado

Trigger: O utilizador não autenticado deseja registrar-se ao sistema.

Pré-condições:

1. Ter acesso a internet.
-

Garantia mínima: O utilizador não autenticado não consegue registrar-se no sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O utilizador não autenticado consegue registrar-se com sucesso no sistema.

Cenário principal de sucesso:

1. O utilizador preenche o formulário de registo.
 2. O sistema valida as informações do utilizador.
 3. O utilizador cria um novo utilizador no sistema baseado nas informações fornecidas e envia um e-mail de confirmação.
 4. O utilizador é redirecionado para a página de login da aplicação.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.

2.b: O sistema deteta que o utilizador já se encontra registado.

1. O sistema deverá apresentar uma mensagem de aviso.

2.c: O sistema deteta que o formulário tem campos em falta.

1. O sistema deverá apresentar uma mensagem de aviso.

2. O sistema deverá destacar os campos em falta.

Tabela 2: Use Case 1.2 - Registrar

UC 1.3 - Recuperar password
Nível de Abstração: Objetivo de utilizador (Fish)
Scope: Uso do sistema
Ator: Utilizador não autenticado
Trigger: O utilizador não autenticado deseja recuperar a sua password.
Pré-condições: 1. Ter acesso a internet.
Garantia mínima: O utilizador não autenticado não consegue recuperar a password, aparecendo uma mensagem sobre o possível problema.
Garantia de sucesso: O utilizador não autenticado consegue recuperar a sua password com sucesso no sistema.
Cenário principal de sucesso: 1. O utilizador preenche o formulário de recuperação. 2. O sistema valida as informações do utilizador. 3. O utilizador envia um link para o e-mail pessoal para confirmação do pedido. 4. O utilizador é notificado que um email foi enviado para validar o pedido. 5. O utilizador confirma o pedido. 6. O sistema altera a password.
Extensões: 2.a: O sistema apresenta problemas no processo. 1. O sistema deverá apresentar uma mensagem de erro. 4.a: O utilizador não recebe o email de validação. 1. O sistema deverá permitir reenviar o email de confirmação. 6.a: Problema com alteração da password. 1. O sistema deverá informar que a sua password não foi alterada. Deverá sugerir que tente mais tarde.

Tabela 3: Use Case 1.3 - Recuperar password

UC 2.1 - Adicionar um novo utilizador

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja adicionar um novo utilizador ao sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue adicionar um novo utilizador ao sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue adicionar um novo utilizador ao sistema.

Cenário principal de sucesso:

1. O utilizador preenche o formulário de inserção de novos utilizadores.
 2. O sistema valida as informações do formulário.
 3. O sistema adiciona um novo utilizador ao sistema, associando-o ao grupo definido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.

2.b: O utilizador já se encontra no sistema.

1. O sistema deverá apresentar uma mensagem de erro e avisar que existe um utilizador no sistema com essa informação.

3.a: Problema com a inserção.

1. O sistema devere informação que a inserção não foi realizada. Deverá sugerir que tente novamente.

Tabela 4: Use Case 2.1 - Adicionar um novo utilizador

UC 2.2 - Atribuir grupo a um utilizador
Nível de Abstração: Objetivo de utilizador (Fish)
Scope: Uso do sistema
Ator: Gestor FireLoc
Trigger: O gestor deseja atribuir um grupo (permissões) a um utilizador existente.
Pré-condições: <ol style="list-style-type: none"> 1. Ter acesso a internet. 2. Estar autenticado no sistema. 3. Ter permissões (Gestor FireLoc).
Garantia mínima: O gestor não consegue atribuir um grupo ao utilizador selecionado, aparecendo uma mensagem sobre o possível problema.
Garantia de sucesso: O gestor consegue atribuir, com sucesso, um grupo ao utilizador selecionado.
Cenário principal de sucesso: <ol style="list-style-type: none"> 1. O gestor seleciona um utilizador. 2. O gestor atribui um grupo ao utilizador selecionado. 3. O sistema valida a atribuição do gestor. 4. O sistema atribui permissões ao utilizador através do grupo associado.
Extensões: <ol style="list-style-type: none"> 1.a: Problema na listagem de utilizadores. <ol style="list-style-type: none"> 1. O sistema deverá apresentar uma mensagem de erro. 2.a: O sistema apresenta problemas no processo. <ol style="list-style-type: none"> 1. O sistema deverá apresentar uma mensagem de erro. 3.a: Problema na validação da ação. <ol style="list-style-type: none"> 1. O sistema deverá apresentar uma mensagem de erro. 4.a: Problema na atribuição de permissões. <ol style="list-style-type: none"> 1. O sistema deverá apresentar uma mensagem de erro. 2. O sistema não deverá atribuir o grupo especificado pelo gestor.

Tabela 5: Use Case 2.2 - Atribuir grupo a um utilizador

UC 2.3 - Atualizar utilizador

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja atualizar a informação de um utilizador.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue atualizar a informação do utilizador selecionado, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue atualizar, com sucesso, a informação do utilizador selecionado.

Cenário principal de sucesso:

1. O gestor seleciona um utilizador.
 2. O gestor preenche o formulário com a nova informação.
 3. O sistema valida os dados inseridos pelo gestor.
 4. O sistema atribui atualiza a informação do utilizador.
-

Extensões:

1.a: Problema na listagem de utilizadores.

1. O sistema deverá apresentar uma mensagem de erro.

3.a: O sistema deteta que o formulário tem campos em falta.

1. O sistema deverá apresentar uma mensagem de erro.

2. O sistema deverá destacar os campos em falta.

4.a: Problema na atualização da informação.

1. O sistema deverá apresentar uma mensagem de erro.

2. O sistema não deverá atualizar a informação do utilizador com a nova informação inserida pelo gestor.

Tabela 6: Use Case 2.3 - Atualizar utilizador

UC 2.4 - Excluir utilizador

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja excluir um utilizador do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue excluir um utilizador do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue excluir, com sucesso, um utilizador do sistema.

Cenário principal de sucesso:

1. O gestor seleciona um utilizador.
 2. O gestor exclui o utilizador.
 3. O sistema processa o pedido do gestor.
 4. O sistema exclui o utilizador do sistema. Isto não implica que a informação por ele disponibilizada seja comprometida.
-

Extensões:

- 1.a: Problema na listagem de utilizadores.
1. O sistema deverá apresentar uma mensagem de erro.
- 3.a: O sistema apresenta problemas no processo.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá continuar com o processo.
- 4.a: Problema ao excluir o utilizador.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá excluir o utilizador.

Tabela 7: Use Case 2.4 - Excluir utilizador

UC 2.5 - Listar utilizadores

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja listar todos os utilizadores do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue listar os utilizador do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue listar, com sucesso, todos os utilizador do sistema.

Cenário principal de sucesso:

1. O gestor lista todos os utilizadores do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema lista todos os utilizadores do sistema.
-

Extensões:

- 2.a: O sistema apresenta problemas no processo.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá continuar com o processo.
- 3.a: Problema ao listar todos os utilizadores.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá listar os utilizadores.
-

Tabela 8: Use Case 2.5 - Listar utilizadores

UC 2.6 - Obter um utilizador específico

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja obter informação sobre um utilizador em específico.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue obter informação sobre um utilizador em específico, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue, com sucesso, obter informação sobre um utilizador em específico.

Cenário principal de sucesso:

1. O gestor pede informação sobre um utilizador específico.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve informação sobre um utilizador em específico..
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao devolver a informação sobre o utilizador.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá devolver informação sobre o utilizador.
-

Tabela 9: Use Case 2.6 - Obter um utilizador específico

UC 3.1 -Listar os grupos de utilizador

Nível de Abstração: Objetivo de utilizador (Fish)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja listar os grupos de utilizador do sistema.

Pré-condições:

1. Ter acesso a internet.
2. Estar autenticado no sistema.
3. Ter permissões (Gestor FireLoc).

Garantia mínima: O gestor não consegue listar todos os grupos do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue listar, com sucesso, todos os grupos de utilizador do sistema.

Cenário principal de sucesso:

1. O gestor lista todos os grupos de utilizador do sistema.
2. O sistema processa o pedido do gestor.
3. O sistema lista todos os grupos de utilizador do sistema.

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao listar todos os grupos de utilizador do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá listar todos os grupos de utilizador do sistema.

Tabela 10: Use Case 3.1 -Listar os grupos de utilizador

UC 3.2 - Adicionar um novo grupo

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja adicionar um novo grupo de utilizador ao sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue adicionar um novo grupo sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue adicionar, com sucesso, um novo grupo de utilizador ao sistema.

Cenário principal de sucesso:

1. O gestor preenche o formulário para adicionar um novo grupo de utilizador ao sistema.
 2. O gestor associa as permissões de utilizador ao grupo.
 3. O sistema processa o pedido do gestor.
 4. O sistema adiciona o novo grupo ao sistema.
-

Extensões:

- 1.a: O sistema deteta que existem campos em falta.
1. O sistema deverá apresentar uma mensagem de aviso.
 2. O sistema deverá destacar os campos em falta.
- 3.a: O sistema apresenta problemas no processo.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá continuar com o processo.
- 4.a: Problema ao adicionar o novo grupo ao sistema.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá adicionar o novo grupo ao sistema.
-

Tabela 11: Use Case 3.2 - Adicionar um novo grupo

UC 3.3 - Atualizar grupo

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja atualizar um grupo de utilizador do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue atualizar um grupo do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue atualizar, com sucesso, um grupo do sistema

Cenário principal de sucesso:

1. O gestor preenche o formulário para atualizar um grupo do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema atualiza o grupo do sistema.
-

Extensões:

1.a: O sistema deteta que existem campos em falta.

1. O sistema deverá apresentar uma mensagem de aviso.
2. O sistema deverá destacar os campos em falta.

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao atualizar um grupo do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá atualizar o grupo do sistema.
-

Tabela 12: Use Case 3.3 - Atualizar grupo

UC 3.4 - Excluir grupo

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja excluir um grupo de utilizador do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue excluir um grupo do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue excluir, com sucesso, um grupo do sistema. Isto implica que os utilizadores que pertenciam a este grupo sejam atualizados e passem a ter permissões default (Grupo de utilizador por defeito - Just a User)

Cenário principal de sucesso:

1. O gestor preenche seleciona o grupo que quer excluir do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema exclui o grupo do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao excluir um grupo do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá excluir o grupo do sistema.
-

Tabela 13: Use Case 3.4 - Excluir grupo

UC 4.1 - Lista contribuições voluntárias

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja listar todas as contribuições do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue listar todas as contribuições do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue listar, com sucesso, todas as contribuições do sistema.

Cenário principal de sucesso:

1. O gestor lista todas as contribuições do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema lista todas as contribuições do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao listar todas as contribuições do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá listar todas as contribuições do sistema.

Tabela 14: Use Case 4.1 - Lista contribuições voluntárias

UC 4.2 - Representar contribuições voluntárias

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja representar todas as contribuições do sistema no mapa.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue representar todas as contribuições do sistema no mapa, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue representar, com sucesso, todas as contribuições do sistema no mapa.

Cenário principal de sucesso:

1. O gestor representa todas as contribuições do sistema no mapa.
 2. O sistema processa o pedido do gestor.
 3. O sistema renderiza todas as contribuições no mapa.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao renderizar todas as contribuições do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá representar todas as contribuições do sistema no mapa.
-

Tabela 15: Use Case 4.2 - Representar contribuições voluntárias

UC 4.3 - Obter contribuição específica

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja obter uma contribuição específica.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue obter a contribuição específica selecionada, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue obter, com sucesso, a contribuição específica que selecionou.

Cenário principal de sucesso:

1. O gestor seleciona a contribuição que quer obter.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve a contribuição selecionada pelo gestor.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao devolver a contribuição selecionada.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá devolver nenhuma contribuição do sistema.
-

Tabela 16: Use Case 4.3 - Obter contribuição específica

UC 4.4 - Excluir contribuições

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja excluir contribuições do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não excluir as contribuições selecionadas, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue excluir, com sucesso, todas as contribuições selecionadas.

Cenário principal de sucesso:

1. O gestor seleciona as contribuições que quer excluir do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema exclui as contribuições selecionadas pelo gestor. Isto inclui que a informação no mapa seja afetada.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao excluir selecionada.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá excluir nenhuma contribuição do sistema das que foram selecionadas.
 3. O sistema não deverá afetar o mapa que representa as contribuições do sistema.
-

Tabela 17: Use Case 4.4 - Excluir contribuições

UC 4.5 - Adicionar contribuições

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja obter as novas contribuições que o sistema validou.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não obtém novas contribuições, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue obter, com sucesso, as novas contribuições que o sistema validou.

Cenário principal de sucesso:

1. O gestor pede as novas contribuições que o sistema validou.
 2. O sistema processa o pedido do gestor.
 3. O sistema apresenta as novas contribuições.
-

Extensões:

- 2.a: O sistema apresenta problemas no processo.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá continuar com o processo.
- 3.a: Problema ao apresentar as novas contribuições.
1. O sistema deverá apresentar uma mensagem de erro.
-

Tabela 18: Use Case 4.5 - Adicionar contribuições

UC 5.1 - Gerir Eventos Reais (Observados à priori)

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os eventos reais (observados à priori) do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os eventos reais (observados à priori) do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os eventos reais (observados à priori) do sistema.

Cenário principal de sucesso:

1. O gestor gere os eventos reais (observados à priori) do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve a resposta do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao devolver a resposta do pedido.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá realizar nenhuma operação relacionada com os eventos reais (observados à priori).

Tabela 19: Use Case 5.1 - Gerir Eventos Reais (Observados à priori)

UC 5.3 - Gerir Anos

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os anos nos quais ocorreram eventos.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os anos nos quais ocorreram eventos, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os anos nos quais ocorreram eventos.

Cenário principal de sucesso:

1. O gestor gere os anos nos quais ocorreram eventos.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve a resposta do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao devolver a resposta do pedido.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá realizar nenhuma operação relacionada com os anos nos quais ocorreram eventos.

Tabela 20: Use Case 5.3 - Gerir Anos)

UC 5.4 - Gerir Áreas Ardidadas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as áreas ardidadas do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as áreas ardidadas do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as áreas ardidadas do sistema.

Cenário principal de sucesso:

1. O gestor gere as áreas ardidadas do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve a resposta do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao devolver a resposta do pedido.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá realizar nenhuma operação relacionada com as áreas ardidadas do sistema.

Tabela 21: Use Case 5.4 - Gerir Áreas Ardidadas

UC 6.1 - Adicionar uma nova camada geospacial

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja adicionar uma nova camada geospacial ao sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue adicionar uma nova camada geospacial ao sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue adicionar, com sucesso, uma nova camada geospacial ao sistema.

Cenário principal de sucesso:

1. O gestor adiciona uma nova camada geospacial ao sistema
 2. O sistema processa o pedido do gestor.
 3. O sistema adiciona uma nova camada geospacial ao sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao adicionar uma nova camada geospacial ao sistema.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá adicionar uma nova camada geospacial ao sistema.

Tabela 22: Use Case 6.1 - Adicionar uma nova camada geospacial

UC 6.2 - Atualizar uma camada geospacial

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja atualizar uma camada geospacial do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue atualizar uma camada geospacial do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue atualizar, com sucesso, uma camada geospacial do sistema.

Cenário principal de sucesso:

1. O gestor atualiza uma camada geospacial do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema atualiza uma camada geospacial do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao atualizar uma camada geospacial do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá atualizar a camada geospacial do sistema.
-

Tabela 23: Use Case 6.2 - Atualizar uma camada geospacial

UC 6.3 - Excluir uma camada geospacial

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja excluir camadas geospaciais do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue excluir as camadas geospaciais do sistema selecionadas, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue excluir, com sucesso, as camadas geospaciais selecionadas.

Cenário principal de sucesso:

1. O gestor exclui as camadas geospaciais selecionadas.
 2. O sistema processa o pedido do gestor.
 3. O sistema exclui as camadas geospaciais do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao excluir as camadas geospaciais do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá excluir as camadas geospaciais do sistema.
-

Tabela 24: Use Case 6.3 - Excluir uma camada geospacial

UC 6.4 - Listar camadas geospaciais

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja listar todas as camadas geospaciais do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue listar todas as camadas geospaciais do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue listar, com sucesso, todas as camadas geospaciais.

Cenário principal de sucesso:

1. O gestor lista todas as camadas geospaciais do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve a lista de todas as camadas geospaciais do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao listar as camadas geospaciais do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá listar as camadas geospaciais do sistema.
-

Tabela 25: Use Case 6.4 - Listar camadas geospaciais

UC 6.5 - Obter uma camada geoespacial específica

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja obter uma camada geoespacial específica do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue obter uma camada geoespacial específica do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue obter, com sucesso, uma camada geoespacial específica do sistema.

Cenário principal de sucesso:

1. O gestor obtém uma camada geoespacial específica do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema devolve uma camada geoespacial específica do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao devolver uma camada geoespacial específica do sistema

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá devolver a camada geoespacial do sistema.

Tabela 26: Use Case 6.5 - Obter uma camada geoespacial específica

UC 6.6 - Atribuir permissões de visualização

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja atribuir permissões de visualização às camadas geospaciais do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue atribuir permissões de visualização às camadas geospaciais do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue atribuir, com sucesso, permissões de visualização às camadas geospaciais do sistema selecionadas.

Cenário principal de sucesso:

1. O gestor atribui permissões de visualização às camadas geospaciais selecionadas.
 2. O sistema processa o pedido do gestor.
 3. O sistema atribui permissões de visualização às camadas geospaciais selecionadas.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema na atribuição de permissões.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá atribuir permissões às camadas geospaciais selecionadas.

Tabela 27: Use Case 6.6 - Atribuir permissões de visualização

UC 6.7 - Gerir estilos

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os estilos de visualização das camadas geospaciais do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os estilos de visualização das camadas geospaciais do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os estilos de visualização das camadas geospaciais do sistema selecionadas.

Cenário principal de sucesso:

1. O gestor gere os estilos de visualização das camadas geospaciais do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema atribui os estilos de visualização das camadas geospaciais do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema na gestão de estilos.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá atribuir estilos às camadas geospaciais selecionadas.
-

Tabela 28: Use Case 6.7 - Gerir estilos

UC 6.8 - Gerir Legendas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as legendas das camadas geospaciais do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as legendas das camadas geospaciais do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as legendas das camadas geospaciais do sistema.

Cenário principal de sucesso:

1. O gestor gere as legendas das camadas geospaciais do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos das legendas das camadas geospaciais do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema na gestão de legendas.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá atribuir legendas às camadas geospaciais selecionadas.
-

Tabela 29: Use Case 6.8 - Gerir Legendas

UC 7.1 - Adicionar um novo gráfico geospacial

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja adicionar um novo gráfico geospacial ao sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue adicionar um novo gráfico geospacial ao sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue adicionar, com sucesso, um novo gráfico geospacial ao sistema.

Cenário principal de sucesso:

1. O gestor adiciona um novo gráfico geospacial ao sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema adiciona um novo gráfico geospacial ao sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao adicionar um novo gráfico geospacial ao sistema.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá adicionar o novo gráfico geospacial ao sistema.

Tabela 30: Use Case 7.1 - Adicionar um novo gráfico geospacial

UC 7.2 - Adicionar dados a um gráfico geospacial

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja adicionar dados a um gráfico geospacial do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue adicionar dados a um gráfico geospacial do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue adicionar, com sucesso, dados a um gráfico geospacial do sistema.

Cenário principal de sucesso:

1. O gestor adiciona dados a um gráfico geospacial do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema associa dados a um gráfico geospacial do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao associar dados a um gráfico geospacial do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve associar dados a um gráfico geospacial do sistema.
-

Tabela 31: Use Case 7.2 - Adicionar dados a um gráfico geospacial

UC 7.3 - Atualizar gráfico geospacial

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja atualizar um gráfico geospacial do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue atualizar um gráfico geospacial do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue atualizar, com sucesso, um gráfico geospacial do sistema.

Cenário principal de sucesso:

1. O gestor atualiza um gráfico geospacial do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema atualiza um gráfico geospacial do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao atualizar o gráfico geospacial do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve atualizar o gráfico geospacial do sistema.
-

Tabela 32: Use Case 7.3 - Atualizar gráfico geospacial

UC 7.4 - Excluir gráficos geospaciais

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja excluir gráficos geospaciais do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue excluir gráficos geospaciais do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue excluir, com sucesso, os gráficos geospaciais do sistema selecionados.

Cenário principal de sucesso:

1. O gestor exclui os gráficos geospaciais do sistema selecionados.
 2. O sistema processa o pedido do gestor.
 3. O sistema exclui os gráficos geospaciais do sistema.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao excluir os gráficos geospaciais do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve excluir os gráficos geospaciais do sistema.
-

Tabela 33: Use Case 7.4 - Excluir gráficos geospaciais

UC 7.5 - Atribuir requisitos de permissões de visualização

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja atribuir os requisitos de permissões a um gráfico geospacial.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue atribuir os requisitos de permissões a um gráfico geospacial, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue atribuir, com sucesso, os requisitos de permissões a um gráfico geospacial.

Cenário principal de sucesso:

1. O gestor atribui os requisitos de permissões a um gráfico geospacial.
 2. O sistema processa o pedido do gestor.
 3. O sistema atribui os requisitos de permissões a um gráfico geospacial.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao atribuir os requisitos de permissões a um gráfico geospacial.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve atribuir os requisitos de permissões a um gráfico geospacial.
-

Tabela 34: Use Case 7.5 - Atribuir requisitos de permissões de visualização

UC 7.6 - Gerir Dados

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os dados de um gráfico geospacial.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os dados de um gráfico geospacial, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os dados de um gráfico geospacial.

Cenário principal de sucesso:

1. O gestor gere os dados de um gráfico geospacial.
 2. O sistema processa o pedido do gestor.
 3. O sistema atribui os dados a um gráfico geospacial.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao atribuir os dados a um gráfico geospacial

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve atribuir os dados a um gráfico geospacial.
-

Tabela 35: Use Case 7.6 - Gerir Dados

UC 8.1 - Gerir Atributos da Avaliação

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os atributos do resultado da avaliação do processo de localização de eventos.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os atributos do resultado da avaliação do processo de localização de eventos, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os atributos do resultado da avaliação do processo de localização de eventos.

Cenário principal de sucesso:

1. O gestor gere os atributos do resultado da avaliação do processo de localização de eventos
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os atributos do resultado da avaliação do processo de localização

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 36: Use Case 8.1 - Gerir Atributos da Avaliação

UC 8.2 - Gerir Resultados da Avaliação

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os resultados avaliação do processo de localização de eventos.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os resultados da avaliação do processo de localização de eventos, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os resultados da avaliação do processo de localização de eventos.

Cenário principal de sucesso:

1. O gestor gere os resultados da avaliação do processo de localização de eventos
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os resultados da avaliação do processo de localização

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 37: Use Case 8.2 - Gerir Resultados da Avaliação

UC 8.3 - Gerir Abordagens da Avaliação

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as abordagens avaliação do processo de localização de eventos.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as abordagens da avaliação do processo de localização de eventos, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as abordagens da avaliação do processo de localização de eventos.

Cenário principal de sucesso:

1. O gestor gere as abordagens da avaliação do processo de localização de eventos
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir as abordagens da avaliação do processo de localização

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 38: Use Case 8.3 - Gerir Abordagens da Avaliação

UC 8.4 - Gerir Procedimentos

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os procedimentos do processo de localização de eventos.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os procedimentos do processo de localização de eventos, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os procedimentos do processo de localização de eventos.

Cenário principal de sucesso:

1. O gestor gere os procedimentos do processo de localização de eventos
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os procedimentos do processo de localização

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 39: Use Case 8.4 - Gerir Procedimentos

UC 9.1 - Gerir Tipos Raster

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os tipos raster do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os tipos raster do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os tipos raster do sistema.

Cenário principal de sucesso:

1. O gestor gere os tipos raster do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os tipos raster do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deve refletir as atividades de gestão realizadas.

Tabela 40: Use Case 9.1 - Gerir Tipos Raster

UC 9.2 - Gerir Datasets Raster

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os datasets raster do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os datasets raster do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os datasets raster do sistema.

Cenário principal de sucesso:

1. O gestor gere os datasets raster do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os datasets raster do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 41: Use Case 9.2 - Gerir Datasets Raster

UC 9.3 - Gerir Camadas Raster

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as camadas raster do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as camadas raster do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as camadas raster do sistema.

Cenário principal de sucesso:

1. O gestor gere as camadas raster do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir as camadas raster do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 42: Use Case 9.3 - Gerir Camadas Raster

UC 9.4 - Adicionar Ficheiros Raster

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja adicionar ficheiros raster ao sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue adicionar ficheiros raster ao sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue adicionar, com sucesso, ficheiros raster ao sistema.

Cenário principal de sucesso:

1. O gestor adiciona ficheiros raster ao sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema adiciona um ficheiro raster.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao adicionar ficheiros raster ao sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve adicionar o ficheiro raster.
-

Tabela 43: Use Case 9.4 - Adicionar Ficheiros Raster

UC 10.1 - Gerir Previsões Meteorológicas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as previsões meteorológicas do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as previsões meteorológicas do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as previsões meteorológicas do sistema.

Cenário principal de sucesso:

1. O gestor gere as previsões meteorológicas do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

- 2.a: O sistema apresenta problemas no processo.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deverá continuar com o processo.
- 3.a: Problema ao gerir previsões meteorológicas do sistema.
1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 44: Use Case 10.1 - Gerir Previsões Meteorológicas

UC 10.2 - Gerir Observações Meteorológicas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as observações meteorológicas do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as observações meteorológicas do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as observações meteorológicas do sistema.

Cenário principal de sucesso:

1. O gestor gere as observações meteorológicas do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir observações meteorológicas do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 45: Use Case 10.2 - Gerir Observações Meteorológicas

UC 10.3 - Gerir Fontes Meteorológicas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as fontes meteorológicas do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as fontes meteorológicas do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as fontes meteorológicas do sistema.

Cenário principal de sucesso:

1. O gestor gere as fontes meteorológicas do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir fontes meteorológicas do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 46: Use Case 10.3 - Gerir Fontes Meteorológicas

UC 10.4 - Gerir estações Meteorológicas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as estações meteorológicas do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as estações meteorológicas do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as estações meteorológicas do sistema.

Cenário principal de sucesso:

1. O gestor gere as estações meteorológicas do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir estações meteorológicas do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 47: Use Case 10.4 - Gerir estações Meteorológicas

UC 10.5 - Gerir Variáveis Meteorológicas

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir as variáveis meteorológicas do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir as variáveis meteorológicas do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, as variáveis meteorológicas do sistema.

Cenário principal de sucesso:

1. O gestor gere as variáveis meteorológicas do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir variáveis meteorológicas do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 48: Use Case 10.5 - Gerir Variáveis Meteorológicas

UC 11.1 - Gerir Atributos da Classificação

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os atributos dos resultados do processo de classificação de fotografias do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os atributos dos resultados do processo de classificação de fotografias do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os atributos dos resultados do processo de classificação de fotografias do sistema.

Cenário principal de sucesso:

1. O gestor gere os atributos dos resultados do processo de classificação de fotografias do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os atributos dos resultados do processo de classificação de fotografias do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 49: Use Case 11.1 - Gerir Atributos da Classificação

UC 11.2 - Gerir Resultados da Classificação

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os resultados do processo de classificação de fotografias do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os resultados do processo de classificação de fotografias do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os resultados do processo de classificação de fotografias do sistema.

Cenário principal de sucesso:

1. O gestor gere os resultados do processo de classificação de fotografias do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os resultados do processo de classificação de fotografias do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deve refletir as atividades de gestão realizadas.

Tabela 50: Use Case 11.2 - Gerir Resultados da Classificação

UC 11.3 - Gerir Procedimentos

Nível de Abstração: Objetivo de utilizador (Sea)

Scope: Uso do sistema

Ator: Gestor FireLoc

Trigger: O gestor deseja gerir os procedimentos do processo de classificação de fotografias do sistema.

Pré-condições:

1. Ter acesso a internet.
 2. Estar autenticado no sistema.
 3. Ter permissões (Gestor FireLoc).
-

Garantia mínima: O gestor não consegue gerir os procedimentos do processo de classificação de fotografias do sistema, aparecendo uma mensagem sobre o possível problema.

Garantia de sucesso: O gestor consegue gerir, com sucesso, os procedimentos do processo de classificação de fotografias do sistema.

Cenário principal de sucesso:

1. O gestor gere os procedimentos do processo de classificação de fotografias do sistema.
 2. O sistema processa o pedido do gestor.
 3. O sistema representa os efeitos do pedido.
-

Extensões:

2.a: O sistema apresenta problemas no processo.

1. O sistema deverá apresentar uma mensagem de erro.
2. O sistema não deverá continuar com o processo.

3.a: Problema ao gerir os procedimentos do processo de classificação de fotografias do sistema.

1. O sistema deverá apresentar uma mensagem de erro.
 2. O sistema não deve refletir as atividades de gestão realizadas.
-

Tabela 51: Use Case 11.3 - Gerir Procedimentos

Apêndice C - Endpoints

/auth/token/{op}/	
POST	Cria/Atualiza o Token de Acesso.
/auth/rqstconfirmation/	
PUT	Envia um email para o utilizador confirmar o registo.
/auth/userconfirmation/	
PUT	Atualiza o token de confirmação do utilizador.
/auth/request-pass-recovery/	
PUT	Envia um email para o utilizador recuperar a sua password.
/auth/pass-recovery/	
PUT	Atualiza a password do utilizador.

Tabela 52: Endpoints do Sistema de Autenticação e Autorização

/auth/users/	
GET	Lista todos os utilizadores do sistema.
POST	Cria um novo utilizador ao sistema (Administrador/Gestor)
DELETE	Apaga todos os utilizadores do sistema.
/auth/user/{userid}/	
GET	Obtém um utilizador específico do sistema.
PUT	Atualiza a informação de um utilizador específico do sistema.
DELETE	Apaga um utilizador específico do sistema.
/auth/justauser/	
POST	Cria um novo utilizador ao sistema (Regular)

Tabela 53: Endpoints da Gestão de Utilizadores

/volu/contributions/	
GET	Lista todas as contribuições dos voluntários.
POST	Regista uma contribuição de um voluntário.
DELETE	Apaga todas as contribuições do sistema.
/volu/contribution/{fid}/	
GET	Obtém uma contribuição específica do sistema.
PUT	Atualiza a informação de uma contribuição específica do sistema.
DELETE	Apaga uma contribuição específica do sistema.
/volu/photo/{picname}/	
GET	Obtém uma fotografia específica de uma contribuição do sistema.

Tabela 54: Endpoints da Gestão de Contribuições

/events/real-fires/	
GET	Lista todas as observações reais de eventos.
POST	Regista uma observação de um evento.
DELETE	Apaga todos as observações reais de eventos do sistema.
/events/real-fire/{fid}/	
GET	Obtém os detalhes de um evento real específico do sistema.
PUT	Atualiza os detalhes de um evento real específico do sistema.
DELETE	Apaga um evento real específico do sistema.

Tabela 55: Endpoints da Gestão de Eventos Reais

/events/years/	
GET	Lista todos os anos em que ocorreram eventos.
POST	Regista um ano em que ocorreu um evento.
DELETE	Apaga todos os anos em que ocorreram eventos.
/events/year/{year}/	
GET	Obtém um ano específico do sistema.
PUT	Atualiza um ano específico do sistema.
DELETE	Apaga um ano específico do sistema.

Tabela 56: Endpoints Gestão de Anos dos Eventos

/events/burned-areas/	
GET	Lista todas as áreas ardidas do sistema.
POST	Regista uma área ardida no sistema.
DELETE	Apaga todos os registos de áreas ardidas.
/events/burned-area/{fid}/	
GET	Obtém os detalhes uma área ardida específica do sistema.
PUT	Atualiza os detalhes uma área ardida específica do sistema.
DELETE	Apaga uma área ardida específica do sistema.

Tabela 57: Endpoints da Gestão de Áreas Ardidas

/floc/fattrs/	
GET	Lista todos os atributos da avaliação de identificação de incêndios.
POST	Regista um atributo da avaliação de identificação de incêndios.
DELETE	Apaga os atributos da avaliação de identificação de incêndios.
/floc/fattr/{slug}/	
GET	Obtém um atributo da avaliação de identificação de incêndios.
PUT	Atualiza um atributo da avaliação de identificação de incêndios.
DELETE	Apaga um atributo da avaliação de identificação de incêndios.

Tabela 58: Endpoints Gestão de Atributos Avaliação de Identificação de Incêndios

/floc/approachs/	
GET	Lista todas as abordagens da avaliação de identificação de incêndios.
POST	Regista uma abordagem da avaliação de identificação de incêndios.
DELETE	Apaga todas as abordagens da avaliação de identificação de incêndios.
/floc/approach/{slug}/	
GET	Obtém uma abordagem da avaliação de identificação de incêndios.
PUT	Atualiza uma abordagem da avaliação de identificação de incêndios.
DELETE	Apaga uma abordagem da avaliação de identificação de incêndios.

Tabela 59: Endpoints Gestão de Abordagens de Avaliação de Identificação de Incêndios

/floc/results/	
GET	Lista todos os resultados da avaliação de identificação de incêndios.
POST	Regista um resultado da avaliação de identificação de incêndios.
DELETE	Apaga todos os resultados da avaliação de identificação de incêndios.
/floc/result/{flocid}/	
GET	Obtém um resultado da avaliação de identificação de incêndios.
PUT	Atualiza um resultado da avaliação de identificação de incêndios.
DELETE	Apaga um resultado da avaliação de identificação de incêndios.

Tabela 60: Endpoints Gestão de Resultados de Avaliação de Identificação de Incêndios

/floc/procedures/	
GET	Lista todos os procedimentos da avaliação de identificação de incêndios.
POST	Regista um procedimento da avaliação de identificação de incêndios.
DELETE	Apaga todos os procedimentos da avaliação de identificação de incêndios.
/floc/procedure/{prid}/	
GET	Obtém um procedimento da avaliação de identificação de incêndios.
PUT	Atualiza um procedimento da avaliação de identificação de incêndios.
DELETE	Apaga um procedimento da avaliação de identificação de incêndios.

Tabela 61: Endpoints Gestão de Procedimentos de Avaliação de Identificação de Incêndios

/floc/photocls-attrs/	
GET	Lista todos os atributos da classificação de fotografias de incêndios.
POST	Regista um atributo da classificação de fotografias de incêndios.
DELETE	Apaga todos os atributos da classificação de fotografias de incêndios.
/floc/photocls-attrs/{slug}/	
GET	Obtém um atributo da classificação de fotografias de incêndios.
PUT	Atualiza um atributo da classificação de fotografias de incêndios.
DELETE	Apaga um atributo da classificação de fotografias de incêndios.

Tabela 62: Endpoints Gestão de Atributos da Classificação de Fotografias de Incêndios

/floc/photos-class/	
GET	Lista todos os resultados da classificação de fotografias de incêndios.
POST	Regista um resultado da classificação de fotografias de incêndios.
DELETE	Apaga todos os resultados da classificação de fotografias de incêndios.
/floc/photo-class/{photoid}/	
GET	Obtém um resultado da classificação de fotografias de incêndios.
PUT	Atualiza um resultado da classificação de fotografias de incêndios.
DELETE	Apaga um resultado da classificação de fotografias de incêndios.

Tabela 63: Endpoints Gestão de Resultados da Classificação de Fotografias de Incêndios

/floc/photo-prcds/	
GET	Lista todos os procedimentos da classificação de fotografias de incêndios.
POST	Regista um procedimento da classificação de fotografias de incêndios.
DELETE	Apaga todos os procedimentos da classificação de fotografias de incêndios.
/floc/photo-prcd/{prid}/	
GET	Obtém um procedimento da classificação de fotografias de incêndios.
PUT	Atualiza um procedimento da classificação de fotografias de incêndios.
DELETE	Apaga um procedimento da classificação de fotografias de incêndios.

Tabela 64: Endpoints Gestão de Procedimentos da Classificação de Fotografias de Incêndios

/georef/ref-grid/	
GET	Lista todas as células da grelha de referência.
POST	Adiciona uma nova célula à grelha de referência.
DELETE	Apaga todas as células da grelha de referência.

Tabela 65: Endpoints Gestão da Grelha de Referência

/georst/raster-types/	
GET	Lista todos os tipos raster do sistema.
POST	Adiciona um novo tipo raster.
DELETE	Apaga todos os tipos raster do sistema.

Tabela 66: Endpoints Gestão de Tipos Raster

/georst/raster-datasets/	
GET	Lista todos os datasets raster do sistema.
POST	Adiciona um dataset raster ao sistema.
DELETE	Apaga todos os datasets raster do sistema.
/georst/raster-dataset/{slugid}/	
GET	Obtém um dataset raster específico do sistema.
PUT	Atualiza um dataset raster específico do sistema.
DELETE	Apaga um dataset raster específico do sistema.

Tabela 67: Endpoints Gestão de Datasets Raster

/georst/ref-grid/	
GET	Lista todos os tipos raster do sistema.
POST	Adiciona um novo tipo raster.
DELETE	Apaga todos os tipos raster do sistema.

Tabela 68: Endpoints Gestão de Tipos Raster

/geovis/charts/	
GET	Lista todos os gráficos geospaciais do sistema.
POST	Adiciona um gráfico geospacial ao sistema.
DELETE	Apaga todos os gráficos geospaciais do sistema.
/geovis/charts/{id}/	
GET	Obtém os detalhes de um gráfico geospacial do sistema.
PUT	Atualiza os detalhes de um gráfico geospacial do sistema.
DELETE	Apaga um gráfico geospacial do sistema.

Tabela 69: Endpoints Gestão de Gráficos

/geovis/chart-data/{chartid}/{serid}/	
GET	Obtém os dados associados a um gráfico geospacial
POST	Associa os dados a um gráfico geospacial.
PUT	Atualiza os dados a um gráfico geospacial.
DELETE	Apaga os dados a um gráfico geospacial.

Tabela 70: Endpoints Gestão de Dados de Gráficos

/geovis/geoportl-layers/	
GET	Lista todas as referências das camadas geospaciais do sistema.
POST	Adiciona uma referência de camada geospacial ao sistema.
DELETE	Apaga todas as referências das camadas geospaciais do sistema.
/geovis/geoportal-layer/{lyr}/	
GET	Obtém os detalhes de uma camada geospacial do sistema.
PUT	Atualiza os detalhes de uma camada geospacial do sistema.
DELETE	Apaga uma camada geospacial do sistema.

Tabela 71: Endpoints Gestão de Camadas

/geovis/layer-groups/{group}	
PUT	Atualiza a relação entre uma camada geospacial e um grupo.
DELETE	Apaga a relação entre uma camada geospacial e um grupo.

Tabela 72: Endpoints Gestão de Permissões das Camadas

/geovis/legends/	
GET	Lista todas as legendas das camadas do sistema.
POST	Adiciona uma nova legenda ao sistema.
DELETE	Apaga todas as legendas das camadas do sistema.
/geovis/legend/{legid}	
GET	Obtém os detalhes de uma legenda do sistema.
PUT	Atualiza os detalhes de uma legenda do sistema.
DELETE	Apaga uma legenda do sistema.

Tabela 73: Endpoints Gestão de Legendas

/meteo/observations/	
GET	Lista todas as observações meteorológicas do sistema.
POST	Adiciona uma observação meteorológica ao sistema.
DELETE	Apaga todas as observações meteorológicas do sistema.
/meteo/observation/{obsid}/	
GET	Obtém os detalhes de uma observação meteorológica do sistema.
PUT	Atualiza os detalhes de uma observação meteorológica do sistema.
DELETE	Apaga uma observação meteorológica do sistema.

Tabela 74: Endpoints Gestão de Observações Meteorológicas

/meteo/forecasts/	
GET	Lista todas as previsões meteorológicas do sistema.
POST	Adiciona uma previsão meteorológica ao sistema.
DELETE	Apaga todas as previsões meteorológicas do sistema.
/meteo/forecast/{fid}/	
GET	Obtém os detalhes de uma previsão meteorológica do sistema.
PUT	Atualiza os detalhes de uma previsão meteorológica do sistema.
DELETE	Apaga uma previsão meteorológica do sistema.

Tabela 75: Endpoints Gestão de Previsões Meteorológicas

/meteo/src/	
GET	Lista todas as fontes meteorológicas do sistema.
POST	Adiciona uma fonte meteorológica ao sistema.
DELETE	Apaga todas as fontes meteorológicas do sistema.
/meteo/src/{slug}/	
GET	Obtém os detalhes de uma fonte meteorológica do sistema.
PUT	Atualiza os detalhes de uma fonte meteorológica do sistema.
DELETE	Apaga uma fonte meteorológica do sistema.

Tabela 76: Endpoints Gestão de Fontes Meteorológicas

/meteo/stations/	
GET	Lista todas as estações meteorológicas do sistema.
POST	Adiciona uma estação meteorológica ao sistema.
DELETE	Apaga todas as estações meteorológicas do sistema.
/meteo/station/{idapi}/	
GET	Obtém os detalhes de uma estação meteorológica do sistema.
PUT	Atualiza os detalhes de uma estação meteorológica do sistema.
DELETE	Apaga uma estação meteorológica do sistema.

Tabela 77: Endpoints Gestão de Estações Meteorológicas

/meteo/var/	
GET	Lista todas as variáveis meteorológicas do sistema.
POST	Adiciona uma variável meteorológica ao sistema.
DELETE	Apaga todas as variáveis meteorológicas do sistema.
/meteo/var/{slug}/	
GET	Obtém os detalhes de uma variável meteorológica do sistema.
PUT	Atualiza os detalhes de uma variável meteorológica do sistema.
DELETE	Apaga uma variável meteorológica do sistema.

Tabela 78: Endpoints Gestão de Variáveis Meteorológicas

/sdi/geosvr/workspaces/	
GET	Lista todos os workspaces do GeoServer.
POST	Adiciona um novo workspace do GeoServer.
/sdi/geosvr/workspace/{ws}/	
GET	Obtém os detalhes de um workspace do GeoServer.
DELETE	Apaga um workspace do GeoServer.

Tabela 79: Endpoints Gestão de Workspaces do GeoServer

/sdi/geosvr/{ws}/stores/	
GET	Lista todas as Stores de um Workspace do GeoServer.
POST	Adiciona uma nova Store a um Workspace do GeoServer.
/sdi/geosvr/{ws}/store/{st}/	
GET	Obtém os detalhes de uma Store de um Workspace do GeoServer.
DELETE	Apaga uma Store de um Workspace do GeoServer.

Tabela 80: Endpoints Gestão de Stores do GeoServer

/sdi/geosvr/layers/	
GET	Lista todas as camadas do GeoServer.
POST	Adiciona uma camada para um determinado Workspace e Store do GeoServer.
/sdi/geosvr/layer/{lyr}/	
GET	Obtém os detalhes de uma camada do GeoServer.
PUT	Atualiza os detalhes de uma camada do GeoServer.
DELETE	Apaga uma camada do GeoServer.

Tabela 81: Endpoints Gestão de Camadas do GeoServer

	<code>/sdi/geosvr/wms/{ws}/</code>
GET	Gera um serviço WMS no GeoServer.
	<code>/sdi/geosvr/wfs/{ws}/{lyr}/</code>
GET	Gera um serviço WFS no GeoServer.

Tabela 82: Endpoints Gestão de Serviços do GeoServer

Apêndice D - Resultados dos Testes (Boxplot)

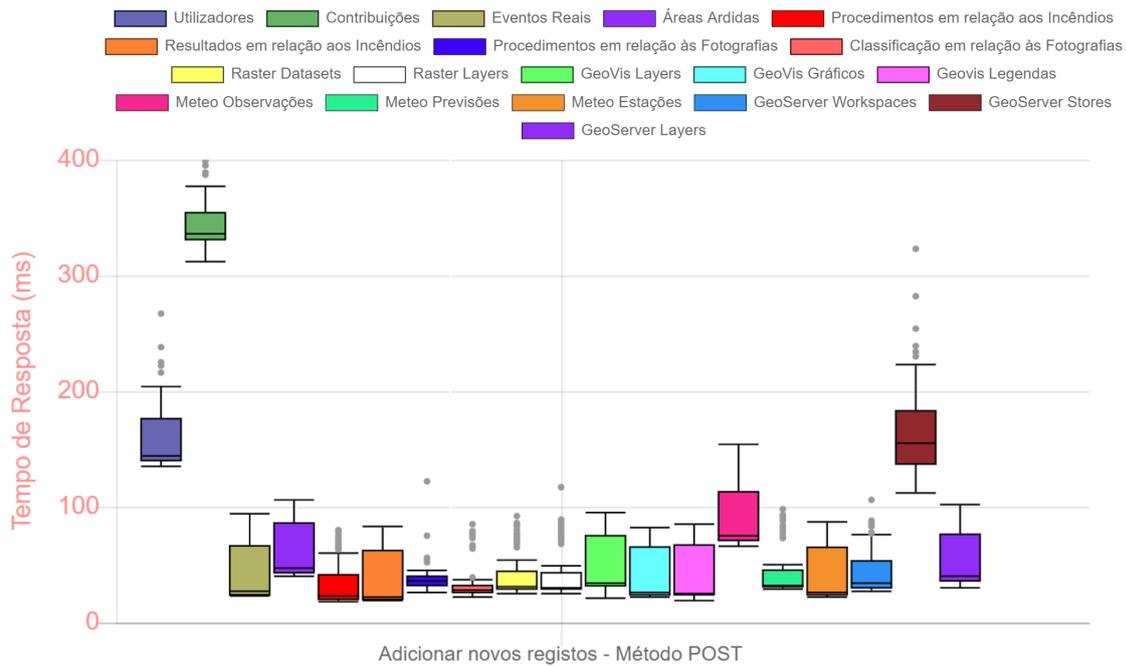


Figura 14: Resultados para o método POST (Adicionar) - Boxplot

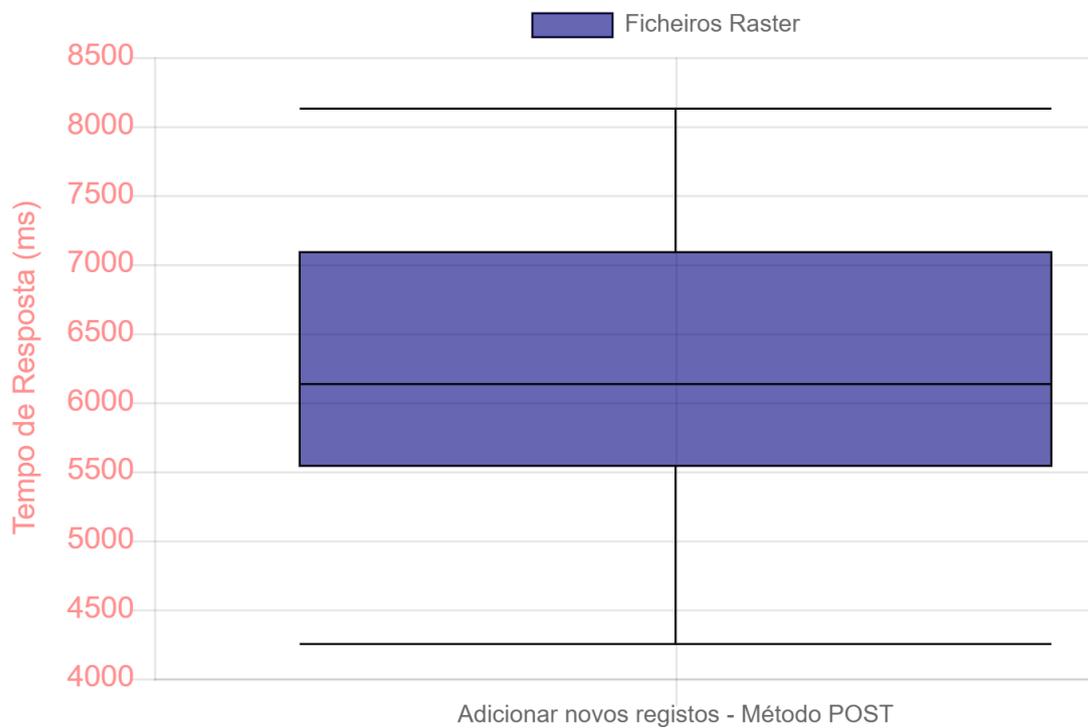


Figura 15: Resultados para o método POST (Adicionar Ficheiros Raster) - Boxplot