



Yuri Raphael da Silva Moreira

EXPANSÃO DO CRISTIANISMO AO LONGO DOS TEMPOS E SUA COMPONENTE GEOESPACIAL EXPRESSA NUM WebSIG

Dissertação no âmbito do Mestrado em Tecnologias de Informação Geográfica, orientada pelo Professor Doutor José Paulo Elvas Duarte de Almeida e pelo Mestre Joaquim António Saraiva Patriarca apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

junho/2022

• U •



UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologia
Departamento de Matemática

EXPANSÃO DO CRISTIANISMO AO LONGO DOS TEMPOS E SUA COMPONENTE GEOESPACIAL EXPRESSA NUM WebSIG

Yuri Raphael da Silva Moreira

Dissertação no âmbito do Mestrado em Tecnologias de Informação Geográfica, orientada pelo Professor Doutor José Paulo Elvas Duarte de Almeida e pelo Mestre Joaquim António Saraiva Patriarca apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Junho de 2022



UNIVERSIDADE DE COIMBRA

Ficha Técnica

Tipo de trabalho	Dissertação de mestrado
Título	Expansão do cristianismo ao longo dos tempos e sua componente geoespacial expressa num WebSIG
Autor	Yuri Raphael da Silva Moreira
Orientador/a(s)	Professor Doutor José Paulo Elvas Duarte de Almeida Mestre Joaquim António Saraiva Patriarca
Júri	Presidente: Doutor João Manuel de Moraes Barros Fernandes Vogais: 1. Doutor Alberto Jorge Lebre Cardoso 2. Doutor José Paulo Elvas Duarte de Almeida
Identificação do Curso	2º Ciclo em Tecnologias de Informação Geográfica
Área científica	Tecnologias de Informação Geográfica
Especialidade/Ramo	Ciências e Tecnologias de Informação Geográfica
Data da defesa	20-07-2022
Classificação	18 valores



UNIVERSIDADE DE COIMBRA

Dedicatória

À minha esposa, Marilena Moreira, que tem me acompanhado desde o início desta trajetória, me apoiando com palavras de incentivos nos momentos cruciais e me motivando nos dias mais difíceis. Apoio imprescindível para realização desse sonho.

À minha filha Catarina, que veio para alegrar nossas vidas, e se tornou minha grande inspiração.

À minha irmã, Polliany Moreira, que diretamente influenciou para que conseguisse chegar até aqui.

Agradecimentos

À Deus, pelo seu cuidado e por suas bênçãos sem medidas sobre minha vida. Sem Ele, nada disso seria possível. Ele é a fortaleza na qual refugio-me. Dele provém toda sabedoria e conhecimento. "Pois o Senhor é quem dá sabedoria; de sua boca procedem o conhecimento e o discernimento." Provérbios 2:6.

À toda a minha família que sempre me apoiou, em especial a minha avó Jane, a minha mãe Cristiany e ao meu pai Itamar, que são exemplos de perseverança e me ensinaram valores e princípios nos quais me espelho.

Ao meu amigo Loandro (in memoriam) com quem vim a primeira vez à Europa, mostrando que o mundo vai muito além daquilo que imaginamos.

Ao meu tio Erivelton (in memoriam) que me encorajou quando era mais novo a seguir o caminho dos estudos.

Ao Professor Doutor José-Paulo, orientador principal do trabalho que desde o princípio auxiliou-me nas mais diversas questões e durante a elaboração trabalho sempre esteve disposto em ajudar no que fosse preciso.

Ao mestre coorientador Joaquim Patriarca, que com suas notáveis competências no que diz respeito às tecnologias de informação geográfica e ao domínio das diversas linguagens de programação, instigou-me à busca por conhecimentos em cada uma das tecnologias utilizadas nesse projeto!

Ao Professor Doutor José Gomes, que no início do curso apoiou-me a seguir adiante com este tema de mestrado.

A todo corpo docente do curso de Mestrado em Tecnologias de Informação Geográfica, em especial à Professora Doutora Cidália Fonte, que dentre as disciplinas lecionadas, a de Modelação e Análise de Dados Geográficos, impulsionou-me para a criação deste projeto, também ao Professor Doutor Alberto Cardoso que transmitiu seus conhecimentos sobre a linguagem de programação Python, fomentando meu interesse por este mundo da programação. O meu muito obrigado a todos pelos conhecimentos compartilhados e por me incentivarem diretamente a chegar até aqui.

À historiadora Gina Zurlo, da World Christian Database, pela atenção e disponibilidade em nos responder e fazer possível a criação deste projeto.

Obrigado a todos!

RESUMO

Observando-se os inúmeros contextos possíveis em que o Sistema de Informação Geográfica pode abranger, e a sua potencialização quando unido com os Sistemas Web, foi proposto o desenvolvimento do WebSIG, designado neste trabalho por Christian World Mapping (CWM), para representar a expansão geoespacial da religião cristã ao longo dos tempos.

Tendo-se verificado que não há nenhum WebSIG semelhante e, dada a relevância destas informações do ponto de vista acadêmico, religioso, histórico e tecnológico, procurou-se desenvolver o CWM com essas características permitindo que estes utilizadores façam as buscas relacionadas com o número de cristãos segundo uma determinada escala territorial e segundo uma determinada escala temporal.

Ancorando-se nos dados alfanuméricos fornecidos pela World Christian Database, instituição responsável por estudos demográficos da religião cristã, foi possível determinar os limites territoriais e o número de cristãos em diferentes intervalos de tempo, desde o surgimento da religião até atualidade e, uma projeção para o futuro.

Para alcançar o objetivo acima, utilizaram-se tecnologias de código aberto para a construção de todo o sistema Web, uma base de dados relacional geográfica para armazenar toda as informações e determinar as relações entre os dados geográficos e os dados alfanuméricos, um servidor REST (*Representational State Transfer*) para a implementação do *backend*, uma aplicação de cliente para a construção do *frontend*, um servidor de mapas para a criação dos diferentes serviços de mapas na Web quando requisitados pelo sistema, e os módulos para pré-processamento de dados e para a geração automática e aplicação de simbologia cartográfica.

Por conseguinte, por meio do uso das tecnologias de código aberto, foi possível criar o CWM com distintas páginas Web. A página com maior interação entre o utilizador e o sistema é constituída por um espaço dedicado à apresentação dos mapas de um lado, e, do outro, os *dropdowns* para a escolha do tipo de território, do ano, do indicador e do território específico, de acordo com a busca requerida pelo utilizador. Após as seleções, surge o mapa permitindo àqueles que a utilizarão uma maior percepção da informação relacionada com o cristianismo.

Portanto, a exemplo do cristianismo, um sistema WebSIG pode ser implementado em diferentes contextos (ordenamento do território, saúde, economia, meio ambiente, agricultura, educação etc.) desde que haja dados geográficos e alfanuméricos para o efeito, possibilitando a disseminação da informação por meio da Web utilizando as tecnologias existentes.

Palavras-chave: Expansão do cristianismo; WebSIG; SIG (Sistemas de Informação Geográfica); Servidor de mapas; Código aberto.

ABSTRACT

Observing the countless possible contexts that can be covered by Geographic Information System (GIS) technologies, along with their potentiation when combined with Web systems, it was proposed the development of a WebGIS, designated in this work by Christian World Mapping (CWM), to document and represent the geospatial expansion of Christianity over time. Having verified that there is no similar WebGIS available from the literature and given the relevance of this information from an academic, religious, historical, and technological point of view, the development of CWM was sought with purposes above enabling users to carry out searches related to the Christian expansion according to certain territorial and/or temporal scales. Anchoring on alphanumeric data provided by the World Christian Database – a USA institution responsible for demographic studies of the Christian religion – it was possible to determine the territorial realms and total number of Christians in different time intervals, ranging from the emergence of the religion to the present, and to work out future projections. To achieve this, open-source technologies were used to build the entire Web system, a geographic relational database to store all the information and determine the relationships between geographic data and alphanumeric data, a REST server (Representational State Transfer) for implementing the backend, a Web client application for building the frontend, a Web map server for creating different map services through the Web when required by the system, and modules for pre-processing of data and for automatic generation and application of cartographic symbology. Therefore, using open-source technologies, it was possible to create CWM consisting of different web pages. The main page, which has implemented within it the greatest interaction between the user and the system, is totally devoted to mapping representations (on one side of the screen) and dropdown menus (on the other side) for choosing: type of territory, year, religion indicator, and specific territory unit, all these in accordance with the search required by the user. After selections above, the respective map comes up on the screen allowing users to have an instant visual perception of both geospatial and temporal extension of Christianity. Despite having been applied to Christianity but without loss of generalisation, a WebGIS system can be implemented in many other contexts (land planning, health, economy, environment, agriculture, education, etc.), enabling the dissemination of relevant information through the Web using currently existing Web 2.0 and GIS technologies.

Keywords: Christianity expansion; WebGIS; GIS (Geographic Information Systems); Map server; Open-source.

ÍNDICE GERAL

Dedicatória.....	IV
Agradecimentos.....	V
Resumo.....	VII
Abstract.....	VIII
Índice Geral.....	IX
Índice de Figuras.....	XI
Índice de Tabelas.....	XII
Acrónimos e Siglas.....	XIII
1. Introdução.....	1
1.1 Contextualização.....	1
1.1.1 Compreendendo o que é religião?.....	2
1.1.2 Uma breve história do judaísmo e do povo judeu.....	4
1.1.3 Uma breve história do cristianismo.....	6
1.2 Motivação.....	10
1.3 Objetivos.....	12
2. Revisão bibliográfica.....	13
2.1 Enquadramento conceptual e tecnológico.....	13
2.1.1 Os Sistemas de Informação Geográfica.....	13
2.1.2 Os Sistemas Web.....	16
2.1.3 Base de dados relacional.....	20
2.1.4 WebSIG.....	21
2.2 Estado da arte.....	22
2.3 A fonte e os dados sobre o crescimento do cristianismo.....	27
3. Arquitetura do sistema, casos de uso e respectivas tecnologias.....	33
3.1 Requisitos funcionais.....	33
3.2 Tecnologias.....	38
3.2.1 PostgreSQL - PostGIS.....	38
3.2.2 Django REST framework.....	39
3.2.3 Angular.....	40
3.2.4 GeoServer.....	41
3.3 Estrutura do sistema.....	41
3.4 Domínio, idioma e <i>mockup</i>	43
4. Implementação do sistema.....	45
4.1 Implementação do Django REST framework.....	45
4.2 Construção da base de dados relacional PostgreSQL PostGIS.....	48
4.3 Manipulação dos dados e criação dos WMS.....	50
4.4 Implementação do Angular.....	54
5. Resultados.....	57
6. Conclusão.....	63
6.1 Trabalho futuro.....	65
6.2 Consideração final.....	66

Bibliografia	67
Anexo A – Mapa das doze tribos de Israel	72
Anexo B – Tabelas com as divisões dos tipos de territórios	73
Anexo C – Apresentação das tabelas com os requisitos funcionais.....	78
Anexo D – Linhas de código exemplificando a manipulação dos dados.....	89

ÍNDICE DE FIGURAS

Figura 1: Fluxograma de como é realizado os serviços na Web. Fonte: Nunes, s.d.	18
Figura 2: Euro Healthy, exemplo de WebSIG.	25
Figura 3: Balança Comercial Portuguesa, exemplo de WebSIG.	26
Figura 4: Georubus (ICNF), exemplo de WebSIG.	26
Figura 5: Exemplo de como vieram os dados da WCD.	29
Figura 6: Exemplo da associação dos países com as “Regiões do mundo”.	30
Figura 7: Utilizador e gestão do utilizador.	34
Figura 8: RF’s que interagem com o utilizador.	34
Figura 9: Gestão dos dados alfanuméricos e geoespaciais.	37
Figura 10: Gestão do GeoServer.	38
Figura 11: Diagrama sobre como a informação percorrerá dentro do sistema do WebSIG. (Adaptação de CINCOVIĆ <i>et al.</i> , 2019).	42
Figura 12: <i>Mockup</i> da página de boas-vindas.	43
Figura 13: <i>Mockup</i> da página principal do WebSIG.	44
Figura 14: Diagrama detalhado sobre a circulação da informação no DRF.	47
Figura 15: Exemplo de código para criação das tabelas, através das classes em Python.	48
Figura 16: Estrutura da base de dados relacional.	50
Figura 17: Exemplo de código, utilizando as linguagens de programação Python e SQL.	53
Figura 18: Estrutura do sistema Christian World Mapping.	56
Figura 19: Página de boas-vindas do Christian World Mapping.	57
Figura 20: Página de login da Christian World Mapping.	58
Figura 21: Página principal do WebSIG Christian World Mapping.	59
Figura 22: Demonstração das informações contidas nos três <i>dropdowns</i> de escolhas obrigatórias.	59
Figura 23: Resultado para os “Continentes”, no ano de 1350 e o indicador o n° de cristão.	60
Figura 24: Resultado para as “Regiões do mundo”, o ano de 400 e indicador o n° de cristão.	60
Figura 25: Resultado para as “Regiões do mundo”, o ano de 1700 e indicador o n° de cristão.	61
Figura 26: Resultado para “Países”, no ano 2020 e o indicador o número de católicos.	62
Figura 27: Resultado para “Países”, no ano 2020 e o indicador o número de protestantes.	62
Figura 28: Resultado para “Países”, no ano 2020 e o indicador o número de ortodoxos.	62
Figura 29: Divisão das doze tribos de Israel.	72

ÍNDICE DE TABELAS

Tabela 1: Diferenças entre as religiões ocidentais e as religiões orientais.....	3
Tabela 2: Principais fatos sobre o povo judeu (de 538 a.C. até 70 d.C.).....	5
Tabela 3: Períodos da igreja cristã.	8
Tabela 4: Os dez princípios para os <i>software open source</i>	15
Tabela 5: Alguns exemplos de WebSIG e suas principais funcionalidades.....	23
Tabela 6: Exemplo de separação dos dados por territórios segundo a WCD.	28
Tabela 7: Distribuição dos dados por territórios, anos e indicadores.	30
Tabela 8: Requisitos funcionais da aplicação, página login e página principal.....	35
Tabela 9: Requisitos funcionais da aplicação, página principal.	36
Tabela 10: Localização de cada tecnologia no servidor.....	42
Tabela 11: Países que compõem as regiões do continente africano.	73
Tabela 12: Países que compõem as regiões do continente americano.	74
Tabela 13: Países que compõem as regiões do continente asiático.	75
Tabela 14: Países que compõem as regiões do continente europeu.	76
Tabela 15: Países que compõem as regiões do continente da Oceania.....	77
Tabela 16: Requisito funcional para a gestão dos utilizadores.	78
Tabela 17: Requisito funcional para a gestão dos tipos de territórios.....	79
Tabela 18: Requisito funcional para a gestão dos anos.	80
Tabela 19: Requisito funcional para a gestão dos indicadores.	81
Tabela 20: Requisito funcional para a gestão dos territórios.	82
Tabela 21: Requisito funcional para a gestão das <i>layers</i>	83
Tabela 22: Requisito funcional para a gestão da população.	84
Tabela 23: Requisito funcional para a gestão dos <i>Workspaces</i> do GeoServer.	85
Tabela 24: Requisito funcional para a gestão dos <i>Stores</i> do GeoServer.....	86
Tabela 25: Requisito funcional para a gestão das <i>Layers</i> do GeoServer.....	87
Tabela 26: Requisito funcional para a gestão dos <i>Styles</i> do GeoServer.	88

ACRÓNIMOS E SIGLAS

API – *Application Programming Interface*
CEGOT – Centro de Estudos de Geografia e Ordenamento do Território
CERN – European Organization for Nuclear Research
CRUD – *Create, Read, Update and Delete*
CSS – *Cascading Style Sheets*
CWM – Christian World Mapping
DRF – Django REST framework
ESRI – Environmental Systems Research Institute
FCTUC – Faculdade de Ciências e Tecnologia da Universidade de Coimbra
FTP – *File Transfer Protocol*
GIS – *Geographic Information System*
HTTP – *HyperText Transfer Protocol*
HTML – *HyperText Markup Language*
INPE – Instituto Nacional de Pesquisas Espaciais do Brasil
INSPIRE – A infraestrutura da União Europeia de informação geográfica.
JPEG – *Joint Photographic Experts Group*
JSON – *JavaScript Object Notation*
MTIG - Mestrado em Tecnologias de Informação Geográfica
OGC – Open Geospatial Consortium
ONU – Organização das Nações Unidas
OSGeo – Open Source Geospatial Foundation
OSI – Open Source Initiative
PNG – *Portable Network Graphics*
REST – *Representational State Transfer*
SIG – Sistemas de Informação Geográfica
SOAP – *Simple Object Access Protocol*
SQL – *Structured Query Language*
SLD – *Styled Layer Descriptor*
SHP – ESRI Shapefile
TWF – The World Factbook
UC – Universidade de Coimbra
URL – *Uniform Resource Locator*
XML – *Extensible Markup Language*
WCD – World Christian Database
WCS – *Web Coverage Service*
WFS – *Web Feature Service*
WGS – *World Geodetic System*
WMS – *Web Map Service*
WWW – *World Wide Web*

I. Introdução

O presente texto documenta o processo de conceptualização, desenvolvimento e implementação de um Sistema de Informação Geográfica na Web que tem como principal finalidade a apresentação e divulgação de informação geográfica e alfanumérica sobre a expansão territorial do cristianismo ao longo dos tempos.

Este primeiro capítulo tem por intuito apresentar uma contextualização sobre o significado de religião, direcionando para as duas religiões monoteístas, judaísmo e cristianismo, falar sobre a motivação que levou o autor à busca de dados relacionados ao crescimento do cristianismo ao longo dos séculos e o empenho para encontrar soluções para o desenvolvimento da aplicação Web, fazer uma abordagem sobre o perfil dos futuros utilizadores do WebSIG, e por fim, apresentar os objetivos gerais deste projeto.

I.1 Contextualização

Em linhas gerais, sempre que há um estudo ou relato sobre um determinado povo, uma comunidade, uma língua, uma região ou uma nação, existe o fator religião a ser considerado, algo que é intrínseco à identidade de uma população. Ao longo dos tempos, certas religiões enraizaram-se em algumas regiões do mundo e, em boa parte dos casos, destaca-se aquela que é predominante, que, por vezes, influencia diretamente o regime político de um estado. Este fato demonstra a importância e influência que a religião tem na sociedade.

Embora existam muitas religiões no mundo com muitos seguidores, o presente estudo dedica-se exclusivamente à maior religião da atualidade, o cristianismo. Objetiva-se, portanto, a construção de um WebSIG que disponibilizará informações sobre a distribuição geográfica mundial do número total e/ou percentagem de cristãos ao longo dos anos. Assim sendo, para além de informações alfanuméricas, o WebSIG terá como uma das suas componentes principais, uma aplicação Web para visualização dos dados atrás identificados sobre a forma de mapa. Esta aplicação será interativa pois permitirá ao utilizador selecionar opções e aplicar filtros geoespaciais e temporais.

Sendo assim, considera-se importante falar sobre o que é uma religião, sobre o judaísmo, raiz do cristianismo, e sobre o próprio cristianismo, abordando principalmente o contexto histórico que leva a justificação de seu crescimento e expansão geoespacial.

1.1.1 Compreendendo o que é religião

Desde os primórdios da civilização humana, o homem, busca entendimento quanto a sua origem e o seu destino após a morte. Tenta compreender os mistérios que circundam a existência, que se permeiam pela natureza e ampliam-se pelo universo.

Durante o intervalo entre o nascimento e a morte, o homem tenta responder: terá sido o homem criado por algo ou alguém que não seja deste mundo? Teria o homem evoluído com o passar dos milhares de anos? Ou teria sido o homem criado e depois evoluído? Com base nestes questionamentos, destacam-se duas principais teorias que visam encontrar respostas, a do criacionismo (através da fé) e a do evolucionismo (através da ciência) (Sampaio, 2006), ambas geram argumentos que não caberiam neste documento.

Portanto, este trabalho terá como ponto de partida apenas a teoria do criacionismo, que para Hodge (1874) citado por Sampaio (2006) em confronto com o evolucionismo faz a analogia dizendo que assim como “o desenho complexo do olho humano só pode ter sido planejado pelo Criador, como o desenho de um relógio demonstra a existência do relojoeiro, portanto a negação do desenho divino na natureza equivale à negação da existência de Deus.”

Neste contexto, crendo no Deus/deuses criador(es), através de quaisquer que sejam as motivações, surge então a religião. Uma palavra que abrange muitos significados e de difícil definição que em termos substantivos e funcionais pode-se considerar de acordo com Coutinho (2012):

“Em termos substantivos, (...) é um sistema composto por descrições do sagrado, respostas ao sentido do mundo e da vida (crenças), meios, sinais, experiências de ligação a esse sagrado (práticas), orientações normativas do comportamento (valores) e atores coletivos com regras e recursos próprios (coletividades).

Em termos funcionais, a religião permite regular e justificar a conduta individual (normativa), providenciar coesão social (coesiva), consolar e aliviar (tranquilizante), fortificar a vontade (estimulante), dar sentido à vida (significante), possibilitar a experiência do sagrado (experiencial), crescer e amadurecer (maturativa), proporcionar identidade (identitária) e ministrar salvação (redentora).”

Para esta palavra considera-se ainda as interpretações dos seguintes autores:

“A religião é um sentimento ou uma sensação de absoluta dependência.” Friedrich Schleiermacher (1768-1834), (citado em Gaarden *et. al.* 1952).

“Religião significa a relação entre o homem e o poder sobre-humano no qual ele acredita ou do qual se sente dependente. Essa relação se expressa em emoções especiais (confiança, medo), conceitos (crença) e ações (culto e ética).” C. P. Tiele (1830-1902), (citado em Gaarden *et al.* 1952).

“A religião é a convicção de que existem poderes transcendentais, pessoais ou impessoais, que atuam no mundo, e se expressa por insight, pensamento, sentimento, intenção e ação. Helmuth von Glasenapp (1891-1963), (citado em Gaarden *et al.* 1952).

Para alguns autores as principais religiões no mundo atual estão divididas em ocidentais e orientais, “consideram-se ocidentais o judaísmo, o islã e o cristianismo, enquanto as principais religiões orientais são o hinduísmo, o budismo e o taoísmo” (Gaarden *et al.*, 1952, p. 41). Segundo destaca Gaarden *et al.* (1952, p. 42), existem algumas diferenças entre as crenças das religiões ocidentais e orientais, conforme observa-se na Tabela 1.

Tabela 1: Diferenças entre as religiões ocidentais e as religiões orientais.

	OCIDENTAL	ORIENTAL
Visão da história	Visão linear da história, isto é, a história tem um começo e um fim; o mundo foi criado num certo ponto e um dia irá terminar.	Visão cíclica da história, isto é, a história se repete num ciclo eterno e o mundo dura de eternidade em eternidade.
Conceito de deus	Deus é o criador; Ele é todo-poderoso e é único. O monoteísmo é tipicamente ocidental.	O divino está presente em tudo. Ele se manifesta em muitas divindades (politeísmo), ou como uma força impessoal que permeia tudo e a todos (panteísmo).
Noção de humanidade	Há um abismo entre Deus e o ser humano, entre o criador e a criatura. O grande pecado é o homem desejar se transformar em Deus em vez de se sujeitar à vontade de Deus.	O homem pode alcançar a união com o divino mediante a iluminação súbita e o conhecimento.
Salvação	Deus redime o ser humano do pecado, julga e dá a punição. Existe a noção de vida após a morte, no céu ou no inferno.	A salvação é se libertar do eterno ciclo da reencarnação da alma e do curso da ação. A graça vem por meio de atos de sacrifício ou do conhecimento místico.
Ética	O fiel é um instrumento da ação divina e deve obedecer à vontade de Deus, abandonando o pecado e a passividade diante do mal.	Os ideais são a passividade e a fuga do mundo.
Culto	Orar, pregar, louvar.	Meditação, sacrifício.

Segundo os dados para o ano de 2020 obtidos no The World FactBook (TWF, 2021), as religiões que têm o maior número de seguidores em percentagem da população mundial são: o

Cristianismo (31,1%), Islamismo (24,9%), Hinduísmo (15,2%), Budismo (6,6%), religiões folclóricas (5,6%), Judaísmo (0,2%), outras (0,8%), não afiliados (15,6%). Os não afiliados são designados pelos ateus, agnósticos ou nada em particular.

A seguir o breve contexto histórico sobre o judaísmo e o cristianismo, que tem suas origens no patriarca Abraão.

1.1.2 Uma breve história do judaísmo e do povo judeu

A primeira religião monoteísta, o judaísmo, surge com o patriarca Abraão cerca de 1800 a.C. na região de Ur, no atual Sul de Iraque (Gaarden *et al*, 1952, pág. 103). O nome da religião deriva de Judéia, antiga região do reino de Israel.

Os judeus têm como base os ensinamentos contidos no livro Tanakh (A Bíblia dos judeus), o livro está dividido em três partes: Torá (A Lei); Neviim (os Profetas); e Ketuvim (os escritos), (Drubi, s.d.).

Em síntese, o surgimento do povo judeu ocorreu, segundo relatos citados no capítulo 12 do livro de Gênesis, escrito pelo profeta Moisés cerca de 1400 a.C., através de Abraão, quando este recebe uma ordem vinda da parte de Deus para que ele partisse para uma nova terra, onde seria formado um novo povo através de sua descendência; o povo que Deus escolhera para Si. É neste momento que Deus estabelece uma aliança com Abraão, conhecida como aliança abraâmica, fortalecida pela promessa de Deus que de um filho, surgiria uma nação.

Abraão e sua esposa Sara, então tiveram o filho prometido por Deus o qual recebeu o nome de Isaque. Isaque e sua esposa Rebeca geraram a Esaú e Jacó. Jacó, por sua vez, teve o seu nome mudado por Deus para Israel e teve 12 filhos e uma filha (Rúben, Simeão, Judá, Levi, Dã, Neftali, Gade, Aser, Issacar, Zebulom, Benjamim, José e Diná). Este pequeno grupo familiar, com cerca de 70 pessoas (Êxodo, 1), dá início ao povo de Israel, também conhecidos por hebreus.

Devido a uma fome muito severa este povo vai em busca de alimentos no Egito e ali vivem como escravos por cerca de 400 anos (Israel, 2021). Deus então promete ao profeta Moisés que libertaria o povo do jugo egípcio e o escolhe como porta-voz do povo para negociar com faraó a sua libertação.

Após a libertação, eles seguiram rumo a terra prometida, Canaã. Durante esta caminhada o profeta Moisés recebeu os 10 mandamentos e informações vindas de Deus determinando como os hebreus deveriam se comportar, conhecida como a Lei mosaica.

A conquista da terra prometida, Canaã (região do atual estado de Israel), foi liderada por Josué. A Terra foi então dividida em doze partes, formando as doze tribos de Israel (Anexo A, Figura 29). Os descendentes de Levi não tiveram parte na divisão da terra, pois eles ficaram responsáveis pelos cuidados sacerdotais no tabernáculo e posteriormente no Templo, locais de culto e adoração à Deus.

No decorrer da história dos judeus, o governo de Israel deixa de ser teocrático e passa a monarquia. O primeiro rei instituído foi Saul, cerca de 1020 a.C. (Israel, 2021), seguido por Davi, cerca de 1004 a 965 a.C. (Israel, 2021), depois de sua morte, Israel passou a ser governada por seu filho Salomão (cerca de 965 a 930 a.C.).

“Após a morte de Salomão (930 AC), uma insurreição aberta levou ao rompimento das dez tribos do norte e à divisão do país num reino do norte, Israel, e um reino do sul, Judá - este último no território das tribos de Judá e Benjamin” (Israel, 2021).

Segundo informações obtidas na página da Embaixada de Israel em Portugal, dando seguimento a história do povo judeu e do judaísmo:

“O Reino de Israel, com a sua capital Samaria, durou mais de 200 anos com 19 reis, enquanto o Reino de Judá foi governado a partir de Jerusalém durante 400 anos pelo mesmo número de reis, da linhagem de David. A expansão dos impérios Assírio e Babilônio causou a dominação de Israel e, depois, de Judá.

O Reino de Israel foi destruído pelos Assírios (722 a.C.) e o seu povo foi levado ao exílio e ao esquecimento. Mais de cem anos depois, a Babilônia conquistou o Reino de Judá, exilando a maioria dos seus habitantes e destruindo Jerusalém e o Templo (586 a.C.)”.

A Tabela 2 mostra os principais acontecimentos do povo judeu desde o exílio na Babilônia (Israel, 2021).

Tabela 2: Principais fatos sobre o povo judeu (de 538 a.C. até 70 d.C.).

PERÍODO	ACONTECIMENTOS
Persa e helenístico (538 a 142 a.C.)	O rei persa Ciro, conquistador do império babilônico (538 a.C.), permite o retorno de cerca de cinquenta mil judeus à Terra de Israel, liderados por Zorobabel; Menos de um século depois, o segundo retorno foi liderado por Esdras, o Escriba; Período em que o Templo é reconstruído; Dentro dos limites do Império Persa, Judá era uma nação liderada pelo sumo sacerdote e conselho de anciãos em Jerusalém; os judeus tiveram diferentes graus de autonomia sob governos Persas (538 a 333 a.C.) e helenísticos (ptolemaico e selêucida) (332 a 142 a.C.). Conquistado por Alexandre, o Grande, da Grécia (332 a.C.), Israel continuou a ser uma teocracia judaica, sob o domínio dos selêucidas.

Tabela 2: Principais fatos sobre o povo judeu (Continuação).

PERÍODO	ACONTECIMENTOS
Dinastia dos Asmoneus (142 a 63 a.C.)	Primeiramente liderados por Matatias, da família sacerdotal dos Asmoneus, e depois por seu filho Judá, os Macabeu a independência judaica foi alcançada. Durante a dinastia dos Asmoneus, que durou aproximadamente 80 anos, o reino recuperou fronteiras quase iguais às do reino de Salomão, alcançou a consolidação política sob o governo judeu e a vida judaica floresceu.
Domínio Romano (63 a.C. a 313 d.C.)	Em 37 a.C., Herodes, foi nomeado rei da Judéia pelos romanos e tornou-se um dos mais poderosos monarcas no Império Romano oriental. Reformou o Templo, tornando-o uma das mais magníficas construções da época. Morreu no ano 4 d.C. Em 70 d.C. Jerusalém foi destruída pelo império romano.

Desde a destruição de Jerusalém até ao século XX a grande maioria dos judeus viveram dispersos pelo mundo, mas mantiveram seus costumes, crenças e tradições. Somente no ano de 1948 Israel se torna um Estado, recebendo o reconhecimento em 1949 pela Organização das Nações Unidas (ONU), (Israel, 2021).

Para estabelecer a ligação com o cristianismo, apresenta-se o seguinte trecho da obra “O Livro das Religiões” Gaarden *et al* (1952):

“Desde a época do exílio babilônico os judeus alimentaram a esperança e a crença de que chegaria um Messias, um novo rei saído da linhagem de Davi. Esse rei ideal iria restabelecer Israel como uma grande potência, e seu povo passaria a viver em eterna felicidade.

Até hoje a expectativa da chegada do Messias continua viva em muitos judeus. Mas nem todos pensam no Messias como uma pessoa; falam, em vez disso, numa futura ‘era messiânica’: um estado de paz na Terra, no qual Israel assumiria um papel de destaque.”

1.1.3 Uma breve história do cristianismo

“Veio para o que era seu, e os seus não o receberam” (Livro de João 1:11).

Com raízes no judaísmo a religião com mais seguidores em todo o mundo, o cristianismo (TWF, 2021), surge a partir da crença de alguns judeus de que Jesus era o então Messias/Cristo prometido e anunciado por Deus através dos antigos profetas judaicos. Messias (hebraico) e Cristo (grego), são palavras idênticas e significam “O Ungido”, o “Príncipe do Reino Celestial” (Hurlbut, 1967).

De acordo com os registos dos evangelhos (os quatro primeiros livros do Novo Testamento da Bíblia), Jesus nasceu na cidade de Belém da Judéia, e cresceu na cidade de Nazaré,

regiões onde está localizado o atual estado de Israel. Neste período o governo era regido pelo imperador romano Tibério César, Pôncio Pilatos era o governador da Judéia e Herodes, o tetrarca da Galileia (Evangelho de Lucas, 3).

Aproximadamente aos 30 anos de idade Jesus começou seu ministério, atraindo seguidores e espalhando a mensagem de arrependimento dos pecados para a salvação eterna. Formou um núcleo de 12 seguidores, que o acompanhou durante 3 anos até a sua morte.

Jesus foi morto pela crucificação, “a pena de morte usada pelos romanos desde o ano 217 a.C. para escravos e todos aqueles que não eram cidadãos do Império” (Ferrara, 2021 citado em Veiga 2021), isto porque os fariseus (grupo de judeus radicais) haviam o denunciado ao governo romano a dizer que Jesus estava a se passar como rei dos judeus e se proclamava o Filho de Deus (uma blasfêmia para os judeus), sendo ele assim, julgado e condenado a morte.

Segundo as escrituras sagradas Jesus ressuscitou ao terceiro dia e, através de aparições em seu corpo ressuscitado manteve contacto direto com seus discípulos durante 40 dias os ordenando a pregação do evangelho, para que as pessoas que ouvissem seus ensinamentos fossem levadas ao arrependimento dos pecados e à crença de que ele é o Filho de Deus e o único caminho para salvação, a vida eterna.

Portanto, “a igreja cristã iniciou sua história com o movimento de caráter mundial, no Dia de Pentecoste, [...], cinquenta dias após a ressurreição de Jesus e dez dias depois de sua ascensão ao céu” (Hurlbut, 1967).

O movimento foi ganhando forças na cidade de Jerusalém, atraindo cada vez mais pessoas naquela região, onde se concentravam. A dispersão que culminou na expansão geográfica, se deu após a morte de Estevão (34 d.C.) apedrejado pelos líderes judeus, e conseqüentemente pelo aumento da perseguição aos cristãos em toda cidade. Entretanto, aonde eles chegassem, seja, Samaria, Damasco ou na longínqua Antioquia, eles pregavam o evangelho e estabeleciam igrejas (Hurlbut, 1967). Foi em Antioquia (atual Antakya, ao Sul da Turquia) que os seguidores de Jesus foram chamados pela primeira vez de cristãos (Atos, 11).

O maior destaque para a expansão inicial do cristianismo se deu através do apóstolo Paulo, judeu, estudioso das escrituras sagradas e que a princípio foi um grande perseguidor aos cristãos, até se converter ao cristianismo, após a aparição de Jesus Cristo enquanto estava a caminho de Damasco. Acreditando que Jesus era realmente o Messias, Paulo começa a pregar o evangelho e implantar novas igrejas em várias cidades através de suas viagens missionárias, ele passa de perseguidor a perseguido. Deixa seu legado através de suas cartas enviadas as igrejas que iniciara. Estas cartas posteriormente viriam a compor boa parte do Novo Testamento da Bíblia Sagrada.

“Sob a direção de Pedro, Paulo e seus sucessores imediatos, a igreja foi estabelecida no espaço de tempo de duas gerações, em quase todos os países, desde o Eufrates até ao Tíbre, desde o Mar Negro até ao Nilo” (Hurlbut, 1967).

Para Hurlbut (1967), a igreja cristã está dividida em 6 períodos, marcados pelos principais acontecimentos citados na Tabela 3.

Tabela 3: Períodos da igreja cristã.

Períodos da Igreja Cristã	Principais acontecimentos
Igreja Primitiva ou Apostólica (30 – 100 d.C.)	<p>Ministério de Jesus; Morte e Ressurreição; Ascensão ao céu;</p> <ul style="list-style-type: none"> - Apóstolos Pedro, João e Tiago como os principais pregadores do evangelho; - Morte de Estevão 34 d.C e perseguições favorecem para a expansão geográfica dos cristãos; - Filipe estabelece igreja em Samaria; - Conversão de Paulo e sua primeira viagem missionária; - Concílio de Jerusalém 50 d.C. define que o evangelho não é somente para judeus, mas sim para todas as pessoas, de todas as raças, e de todas as nações; - Segunda e terceira viagem missionária de Paulo, evangelho chega a Europa; - Em 64 d.C. grande incêndio em Roma, culpa atribuída a Nero que para se defender culpou os cristãos, aumentando as perseguições; - Mártir de Pedro em 67 d.C.; - Mártir de Paulo em 68 d.C., na cidade de Roma; - Destruição de Jerusalém (70 d.C.) pelos romanos liderados por Tito, filho do recém imperador Vespasiano. Foi o marco para a desvinculação do cristianismo do judaísmo, porque até então o cristianismo era considerado por muitos como uma ramificação do judaísmo. - De 70 d.C. até a morte de João 100 d.C. não há muitos registros. Sabe-se que foi durante esse período que se escreveram os últimos livros do Novo Testamento — Hebreus, talvez a Segunda epístola de Pedro, as três epístolas e o evangelho de João, epístola de Judas e o Apocalipse.
Igreja Perseguida (101 – 313 d.C.)	<ul style="list-style-type: none"> - Aumento da perseguição aos cristãos pelos imperadores romanos, devido à: ausência de adoração ao imperador, sendo considerada traição; as reuniões entre os cristãos despertavam suspeitas; as pregações de que todos os homens são iguais; interesses econômicos; - Mortes de cristãos no Coliseu de Roma; - Início da organização eclesiástica, fazendo uma analogia ao governo imperial; - Desenvolvimento da doutrina através da abertura de escolas em Alexandria, Ásia menor e na cidade de Cartago no Norte da África; - Expansão geográfica através do império romano

Tabela 3: Períodos da igreja cristã. (Continuação)

Períodos da Igreja Cristã	Principais acontecimentos
Igreja Imperial (313 – 476 d.C.)	<ul style="list-style-type: none"> - Constantino o Primeiro Imperador Cristão; - 313 édito de Constantino colocando o fim à perseguição e oficializando a religião cristã; - Privilégios concedidos ao clero; - Domingo como o dia de descanso é determinado; - Fundação da cidade de Constantinopla (atual Istambul) por Constantino; - Os Concílios de Nicéia em 325, o de Constantinopla em 381 e o de Cartago em 418. - O nascimento do Monacato; - Queda de Roma em 476; - Principais nomes neste período: Atanásio, (293-373), Ambrósio de Milão, (340-397), João Crisóstomo, (345-407), Jerónimo, (340-420) tradutor da Bíblia para o latim, conhecida como Vulgata, Agostinho, (354-430)
Igreja Medieval (476 – 1453 d.C.)	<ul style="list-style-type: none"> - O poder papal dividido em três épocas: Período de Crescimento, 590-1073 d.C.; Período Culminante, 1073-1216 d.C.; Período de Decadência 1216-1453; - O sacro império romano, fundado por Carlos Magno 742-814; - A separação das igrejas latinas e gregas; - As Cruzadas, 1095-1270 d.C.; - As Ordens Monásticas; - Início da Reforma Religiosa
Igreja Reformada (1453 – 1648 d.C.)	<ul style="list-style-type: none"> - Influências que conduziram à reforma: a renascença, a invenção da imprensa, e o espírito nacionalista; - A venda das indulgências; - As 95 teses de Martinho Lutero; - Reforma da Igreja na Suíça, França, Inglaterra, Países Baixos e Escócia; - A Contrarreforma na igreja Católica. - Expansão do cristianismo no período das Grandes Navegações.
Igreja Moderna (1648 – século XX)	<ul style="list-style-type: none"> - Marcada pelos grandes movimentos: Puritano, o avivamento Wesleyano, o Racionalista, Anglo-Católico e o movimento Missionário; - Expansão do cristianismo nas Américas, o protestantismo avança na América do Norte, quanto o catolicismo para a América Latina.

Portanto, nesta breve pontuação dos fatos que ocorreram ao longo dos anos desde a morte de Jesus, observa-se como a religião que tem suas raízes no judaísmo, passou por inúmeros eventos, se multiplicou em cada período e se espalhou por toda a terra, com predominância no mundo ocidental.

A seguir veremos o que incentivou o autor a relacionar a religião cristã com os Sistemas de Informação Geográfica, um tema pouco usual num contexto tecnológico.

1.2 Motivação

O que levou ao encorajamento para o desenvolvimento deste trabalho foram dois fatores que são considerados como os pilares. O primeiro é o religioso que, após um período de dois anos de estudos no Curso Básico de Teologia, ministrado no Instituto Kletos, localizado na região de Sintra – Portugal, pôde-se aprofundar os conhecimentos relacionados ao cristianismo. Durante esta formação, mais precisamente na disciplina História da Igreja, instigou-se a pensar sobre cada época do cristianismo e a sua expansão geográfica.

O segundo pilar que fomentou ainda mais este desejo de seguir adiante foram, os conhecimentos adquiridos durante as formações do curso em Gestão Ambiental, nas classes de Cartografia e Geoprocessamento e, o curso de Mestrado em Tecnologias de Informação Geográfica, através de suas unidades curriculares. Estas competências, despertaram um enorme gosto pelos SIG, sobretudo a possibilidade de o relacionar com um sistema Web.

Em um período de estudos, numa das aulas da disciplina de Cartografia e WebSIG, no primeiro ano do curso de Mestrado em Tecnologias de Informação Geográfica (MTIG), surgiu o *insight*: “Será que existe algum WebSIG que representa a expansão territorial do cristianismo?”. Mediante esta questão surgiram ainda outras:

- ❖ Como foi este processo de expansão em termos geoespaciais no decorrer dos séculos?
- ❖ Será que existe alguma base de dados onde se encontra o número de cristãos por século e por regiões do mundo?
- ❖ Será que existe algum WebSIG no qual poderia apresentar estas informações de modo interativo?
- ❖ Se não há, seria possível mapear estes dados através de um Sistema de Informação Geográfica (SIG), antes da criação do WebSIG?
- ❖ E para criar este WebSIG, o que será preciso?
- ❖ Quais as capacidades que serão necessárias adquirir?
- ❖ Existe a possibilidade de construir este WebSIG sem custos? Se sim, quais os *software*? Que ferramentas?

Assim, através do levantamento destas questões, instigou-se a procura de informações a respeito do assunto na Web, constatando que realmente não existe um WebSIG disponível em

rede que aborde este tema. Motivando o aprofundamento nas pesquisas para a criação do WebSIG e a busca por conhecimentos técnicos para o desenvolver.

Outro fator que motivou foi encontrar a instituição World Christian Database(WCD), fazer o contacto com os responsáveis e sem muito tempo a espera obter os dados em diferentes escalas geoespacial e temporal relacionados com a população cristã.

O empenho em aprender, tornou-se a principal chave e o impulsionamento necessário para vencer os desafios encontrados, no que diz respeito as linguagens de programação e as tecnologias utilizadas para construção de um WebSIG. Foi preciso formações extracurriculares como: o workshop “Introdução ao desenvolvimento de aplicações e serviços Web sustentados por Informação Geográfica”; a repetição da disciplina Cartografia e WebSIG do curso de M.TIG, para absorver mais conhecimentos e a participação na disciplina de Programação para WebSIG do curso de Mestrado em Engenharia de Informação Geoespacial da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC). Possibilitando uma maior compreensão de como um sistema Web é integrado com os SIG. Portanto, este desejo em aprender mais sobre as tecnologias e poder contribuir de alguma forma com a sociedade, estimulou o desenvolvimento do WebSIG.

Considerando o tema deste projeto, observa-se a potencial contribuição que esta aplicação proporcionará a nível académico, estudantes e professores nos mais diversos cursos como: História geral, Ensino Religioso, Ciência da Religião, Teologia, Histórias das Religiões, Cultura Religiosa, Antropologia da Religião, Sociologia etc. poderão utilizar as informações contidas no WebSIG, tornando-se uma fonte fiável de pesquisas.

Para os cristãos estas informações servirão para a aquisição de conhecimentos para a consolidação da fé cristã, observar o aumento ou diminuição do número de cristãos em cada território e, através disto, compreender, se sensibilizar e serem encorajados ao desenvolvimento de projetos em conjunto à sua comunidade cristã para o envio de missionários afim de evangelizar nestes países com o menor número de cristãos.

Para o utilizador não cristão, que ainda não conhece os fundamentos do cristianismo e sua história, poderá através deste WebSIG, adquirir conhecimentos sobre a religião e observar por onde os cristãos estão concentrados na atualidade, e por onde estiveram.

Desenvolvedores Web, técnicos e analistas em SIG, engenheiros informáticos, e todos aqueles que tenham interesse em aprender sobre um sistema Web e sua junção com os SIG, que queiram adaptar esse projeto a algum outro, poderão fazê-lo através da utilização dos códigos deste trabalho que estarão abertos para o compartilhamento da informação por meio do GitHub.

Portanto o WebSIG, será aberto para todas as pessoas, independentemente de sua faixa etária, sua crença, sua religião e/ou sua etnia, que queiram obter informações relativas ao número de cristãos, em determinado tempo e território e, para aqueles que queiram ter acesso ao código fonte.

1.3 Objetivos

Com o intuito de concretizar a motivação exposta na secção anterior, destacando-se que a intenção não é argumentar ou defender a fé cristã, ou o criacionismo, e sim a disponibilização na Web de informações sobre a história da religião, apresentando o número de cristãos por um determinado território e determinado ano, através da criação de mapas que permitirão o utilizador interagir com a aplicação.

Assim, o objetivo geral do trabalho consiste em desenvolver uma aplicação WebSIG, através da criação de um *backend* e de um *frontend*; construir uma base de dados relacional para armazenar todos os dados alfanuméricos extraídos da WCD e os dados geográficos, possibilitando a criação dos relacionamentos entre estes dois tipos de dados; utilizar um servidor de mapas para a geração de serviços de mapas na Web e, módulos para o pré-processamento dos dados e para a geração automática da simbologia cartográfica. Por último, para não acarretar custos ao projeto, serão utilizadas as tecnologias de código aberto disponíveis para cada um deste ambientes.

Quanto a espacialização desta expansão da religião no decorrer dos anos, por ser uma tarefa de difícil definição em termos geoespaciais, o projeto estará totalmente ancorado nestes dados da WCD que tem as definições dos limites territoriais já estabelecidas, com dados para o “Mundo”, os dados para os “Continentes”, dados para as “Regiões do mundo” e os dados para os “Países”. Em relação ao avanço temporal e aos seus intervalos, também serão utilizadas as informações definidas pela WCD, ou seja, de acordo com os dados adquiridos é que este projeto será desenvolvido. Mais adiante no capítulo 2, há uma secção a explicar com maiores detalhes estes dois aspectos.

2. Revisão bibliográfica

O presente capítulo serve para enquadrar os conceitos e tecnologias fundamentais deste trabalho, nomeadamente: o conceito de Sistemas de Informação Geográfica, Web, base de dados relacionais, serviços de mapas e WebSIG.

Destacam-se também alguns exemplos de WebSIG, que influenciaram diretamente a realização deste projeto.

Ainda neste capítulo, apresenta-se a fonte de dados usada, a WCD, bem como as metodologias utilizadas para a consolidação da referida base de dados. Como complemento, explica-se também como é que os dados foram extraídos e como foram organizados.

2.1 Enquadramento conceptual e tecnológico

Para compreender a funcionalidade de cada tecnologia que será utilizada na construção do WebSIG, essa secção destina-se a explicar os conceitos e termos de cada etapa do desenvolvimento da aplicação, mesmo que seja de forma sucinta, a intenção é deixar claro para o leitor o significado e a aplicabilidade de cada uma das componentes do sistema.

2.1.1 Os Sistemas de Informação Geográfica

Um SIG não é composto apenas pelos *software*, ele envolve muitas outras componentes para que seja possível a representação da realidade mas, antes de falar sobre a composição de um SIG, veremos suas definições.

De acordo com o Instituto Nacional de Pesquisas Espaciais do Brasil (INPE, 2021) a definição para SIG é “um sistema que processa dados gráficos e não gráficos (alfanuméricos) com ênfase a análises espaciais e modelagens de superfícies”. O próprio INPE, cita autores para explicar o que é um SIG, assim, para eles é:

"Um conjunto manual ou computacional de procedimentos utilizados para armazenar e manipular dados georeferenciados" (Aronoff, 1989, citado em INPE, 2021).

"Conjunto poderoso de ferramentas para coletar, armazenar, recuperar, transformar e visualizar dados sobre o mundo real" (Burrough, 1986, citado em INPE, 2021).

"Um sistema de suporte à decisão que integra dados referenciados espacialmente num ambiente de respostas a problemas" (Cowen, 1988, citado em INPE, 2021).

Deste modo, um Sistema de Informação Geográfica (SIG) permite visualizar, questionar, analisar e interpretar dados para compreender relações, padrões e tendências (ESRI Portugal, 2021).

Considera-se uma definição mais completa de um SIG como “um sistema de informação computacional que permite a aquisição, modelação, armazenamento, obtenção, partilha, manipulação, análise e apresentação de dados georreferenciados.” (Worboys & Duckham, 2004, p. 2).

Portanto, um SIG é constituído por diversos mecanismos que irão receber dados como input, e de acordo com o objetivo do trabalho a desenvolver, estes serão transformados em novos dados, de modo que a cartografia seja a representação do resultado destes procedimentos. Um SIG é composto por *hardware*, *software*, dados, recursos humanos, aplicações, procedimentos e rede (Fonte, 2021).

Hardware “engloba a CPU (unidade central de processamento), armazenamento (discos duros, discos ópticos, drives de CD-ROM), dispositivos de entradas de dados (scanners, mesas digitalizadoras) e dispositivos de saída (impressoras, plotters)”, (Cunha, 2009, p.12)

Software são os programas computacionais que trabalham os dados geográficos e/ou alfanuméricos, a fim de produzir informações de acordo com os métodos aplicados. Os *software* são classificados em três principais grupos: proprietário, livre e de código aberto.

O *software* proprietário é aquele que pertence a uma empresa privada e detém seus direitos de uso, edição ou redistribuição sobre ele (Conceitos, 2014). Alguns exemplos aplicados aos SIG: ArcGIS (ESRI), Geomedia (Hexagon Geospatial), MapInfo Professional (Precisely), Bentley Map, AutoCAD Map 3D (Autodesk), IDRISI (Clark Laboratories), etc. (GISGeography, 2021).

O *software* livre os “usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o *software*” (GNU, 2021), existindo quatro princípios para serem considerados livres, estes instituídos pela Free Software Foundation. Embora estes *software* deem a liberdade ao utilizador, isto não significa que eles possam ser comercializados, nem todos os *software* livres são gratuitos. Bohrer (2014), dá alguns exemplos de *software* livre: sistema operacional Linux, o servidor Apache, a base de dados MySQL, o editor de imagens GIMP e o gerenciador de conteúdo na Web WordPress.

Por último, os *software* de código aberto, que seguem os dez princípios definidos pela Open Source Initiative (2007), Tabela 4. De acordo com a organização OSGeo (2022), um

software open source, “começa com uma licença que fornece (re)uso de *software* livre de *royalties*. Garantindo o acesso ao código-fonte para auditoria e modificação e a capacidade de redistribuir o *software* sem custos adicionais.”

Tabela 4: Os dez princípios para os *software open source*.

PRINCÍPIO	DESCRIÇÃO
Redistribuição Livre	A licença não deve restringir nenhuma parte de vender ou dar o <i>software</i> como um componente de uma distribuição de <i>software</i> agregada, contendo programas de várias fontes diferentes.
Código fonte	O programa deve incluir o código-fonte e deve permitir a distribuição no código-fonte, bem como na forma compilada.
Trabalhos derivados	A licença deve permitir modificações e trabalhos derivados, e deve permitir que eles sejam distribuídos sob os mesmos termos da licença do <i>software</i> original.
Integridade da autoria do código fonte	A licença pode restringir o código-fonte de ser distribuído na forma modificada apenas se a licença permitir a distribuição de "arquivos de <i>patch</i> " com o código-fonte com o propósito de modificar o programa no momento da construção. A licença deve permitir explicitamente a distribuição de <i>software</i> construído a partir do código-fonte modificado.
Sem discriminação contra pessoas ou grupos	A licença não deve discriminar nenhuma pessoa ou grupo de pessoas.
Sem discriminação contra campos de atuação	A licença não deve restringir ninguém de fazer uso do programa em um campo específico de atuação.
Distribuição de Licença	Os direitos associados ao programa devem se aplicar a todos a quem o programa é redistribuído, sem a necessidade de execução de uma licença adicional por essas partes.
A licença não deve ser específica para um produto	Os direitos associados ao programa não devem depender do programa ser parte de uma distribuição de <i>software</i> específica.
A licença não deve restringir outro <i>software</i>	A licença não deve impor restrições a outro <i>software</i> distribuído junto com o <i>software</i> licenciado.
A licença deve ser neutra em termos de tecnologia	Nenhuma disposição da licença pode ser baseada em qualquer tecnologia individual ou estilo de interface.

Alguns exemplos de *software* de código aberto voltados para os SIG: QGIS, GRASS GIS, GvSIG, MapServer, OpenLayers, PostGIS, uDig, GeoServer (Software Público, 2021).

Deste modo os *software* têm a função de: armazenar e gerir os dados, através da base de dados; transformar estes dados por meio de operações de análise; interagir com o utilizador pelo acesso aos menus ou *scripts* de programação; permitir uma saída com as novas informações (Fonte, 2021).

Os dados sem dúvida são uma importante componente de um SIG, sem dados não há novas informações. Eles podem ser obtidos gratuitamente ou não, por vezes podem ser dispendiosos. Estão separados em dados geoespaciais, que envolve uma posição no espaço geográfico que por sua vez, envolve a componente planimétrica e altimétrica, deste modo, são quaisquer dados com uma referência direta ou indireta a uma localização ou zona geográfica específica (Diretiva Inspire); e os dados alfanuméricos, que são associados aos dados geoespaciais, por meio de uma base de dados. Existem dois tipos de dados geoespaciais que são introduzidos nos SIG, os vetoriais, definidos pelas formas geométricas: pontos, linhas e polígonos, e os dados *raster*/matriciais, “constituída por uma discretização do espaço em células dispostas de forma regular cuja posição é identificável por índice de linha e coluna, em conjunto com a coordenada da primeira célula e com a dimensão das células (Cunha, 2009, p.14).

Os recursos humanos “constituem a componente fundamental para lidar com os procedimentos complexos desta tecnologia, desde o levantamento, armazenamento, tratamento, apresentação e análise de dados” (Cunha, 2009, p.12). Os SIG precisam de pessoas qualificadas para conceber, construir, manter e interpretar os resultados (Fonte, 2021).

Conclui-se que, os SIG podem ser aplicados em diversos setores: ambiente, agricultura, ordenamento do território, saúde, serviços públicos, transporte, censo, cultura, turismo e património etc. Sendo a concepção do SIG sempre dependente de sua finalidade. Após sua criação, o SIG deverá ser monitorizado para uma possível atualização, análise ou controlo.

2.1.2 Os Sistemas Web

Antes de compreender o que é e como funciona o WebSIG, é importante entender o conceito de Web. De acordo com Nunes (s.d.), a “World Wide Web (WWW ou Web) é um sistema hipertexto que funciona sobre a Internet. A visualização da informação e navegação é feita usando uma aplicação específica - o navegador (browser)”.

A Web começou a ser desenvolvida em dezembro de 1990, para facilitar o compartilhamento de conhecimento, entre um grupo de pesquisadores da European Organization for Nuclear Research (CERN), em Geneva, Suíça. Foi Tim Berners-Lee quem iniciou um projeto de *software* sem fins lucrativos intitulado por “WorldWideWeb” (Massé, 2012).

O sistema Web é baseado em cliente-servidor, sendo cliente designado neste trabalho por *frontend* e o servidor por *backend*, no qual clientes e servidores possuem partes distintas a serem desenvolvidas (Massé, 2012, p.3). Ambos podem ser implementados e desenvolvidos de forma independente, usando diferentes linguagens de programação e diferentes tecnologias (Massé, 2012, p.3). O *frontend* é onde define-se a estrutura semântica do conteúdo, os estilos a serem utilizados e adiciona a interação com o utilizador (Luna *et al.*, 2018), neste lado do sistema Web que são realizados os pedidos ao servidor. Quanto ao *backend*, ele é responsável pela gestão dos conteúdos tendo as principais funções, de acordo com Gourley *et al.* (2002, p.113):

1. Configurar a conexão — aceita uma conexão de cliente ou fecha se o cliente não for desejado.
2. Receber solicitação — ler uma mensagem de solicitação HTTP (*HyperText Transfer Protocol*) da rede.
3. Processar solicitação — interpretar a mensagem de solicitação e tome medidas.
4. Aceder recurso — aceder o recurso especificado na mensagem.
5. Construir resposta — criar a mensagem de resposta HTTP com os cabeçalhos corretos.
6. Enviar resposta — enviar a resposta de volta ao cliente.
7. Transação de log—coloque notas sobre a transação concluída em um arquivo de log.

Deste modo o utilizador faz um pedido indicando a URL¹ (*Uniform Resource Locator*), que solicita ao servidor por meio do HTTP (transporte), este devolve o conteúdo solicitado em formato HTML (*HyperText Markup Language*), permitindo ao utilizador uma iteração com o sistema (Figura 1). Vale ressaltar que existem outros protocolos para a transmissão do conteúdo para além do HTTP mas, este é o mais conhecido e utilizado pela conversa bidirecional entre navegadores da Web e servidores da Web (Gourley *et al.* 2002). Ressalta-se ainda que, existem outros formatos de ficheiros que podem ser devolvidos pelos servidores para além do HTML embora, seja este o mais utilizado.

¹URL - “descrevem a localização específica de um recurso em um determinado servidor. Eles dizem exatamente como buscar um recurso em um local preciso e fixo” (Gourley *et al.* 2002, p.6).

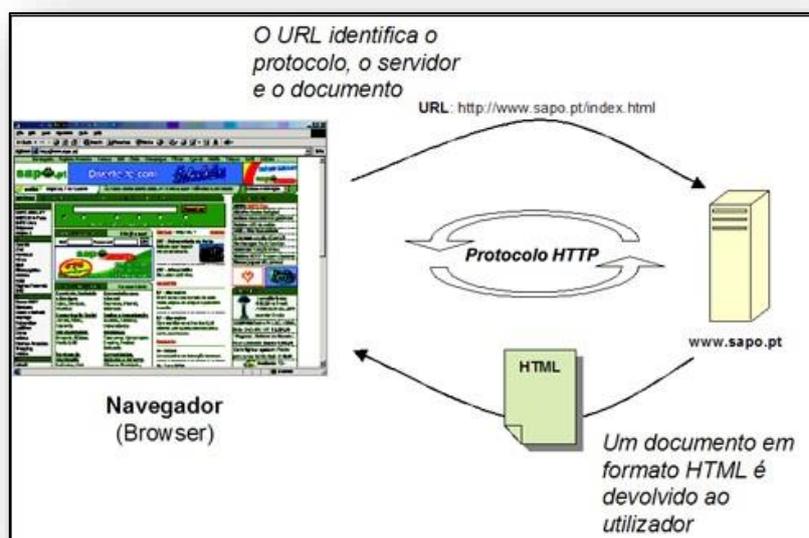


Figura 1: Fluxograma de como é realizado os serviços na Web. Fonte: Nunes, s.d.

Para clarificar “cada mensagem de solicitação HTTP tem um método. O método informa ao *backend* qual ação executar (buscar uma página da Web, executar um programa, excluir um arquivo etc.)” (Gourley *et al.* 2002, p.8). Assim, o “protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso” (Mozilla, s.d.). Destacando-se os principais métodos:

- ❖ GET– solicita a representação de um recurso específico, estas requisições devem retornar apenas dados.
- ❖ POST – “é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor” (Mozilla, s.d.).
- ❖ PUT – “substitui todas as atuais representações do recurso de destino pela carga de dados da requisição” (Mozilla, s.d.). Faz a atualização do registo.
- ❖ DELETE– elimina um recurso específico.

Salienta-se que existe a diferença entre um *Web Site* de um *Web Service*, segundo a Federal Geography Data Committee (FGDC, 2010), um *Web Site* fornece páginas e formulários HTML para usuários navegarem e executarem funções: Pesquisas, Compras, Interação. Utilizando interfaces de usuário *frontend* através do navegador.

Um *Web Service* por sua vez é “utilizado para transferir dados através de protocolos de comunicação para diferentes plataformas, independentemente das linguagens de programação utilizadas nessas plataformas” (Opensoft, 2016). Transmitindo apenas informações entre os sistemas, não permitindo o acesso pelo utilizador através dos navegadores Web. Por trabalhar com as mais diversas linguagens de programação, o *Web Service* necessita de “linguagem intermédia que garanta a comunicação entre a linguagem do Web Service e o sistema que faz o pedido ao Web Service”. (Opensoft, 2016). Para isso, existem os dois principais estilos de arquitetura da informação, o SOAP (*Simple Object Access Protocol*) que utiliza o formato de ficheiro XML (*Extensible Markup Language*) e o REST (*Representational State Transfer*), que utiliza vários formatos de ficheiros para a representação de dados, o mais usual é o JSON (*JavaScript Object Notation*). O REST é um estilo de arquitetura mais recente, mais flexível por operar com vários formatos e mais ágil, com a capacidade de transmitir dados diretamente via protocolo HTTP (Opensoft, 2016). Portanto, o *Web Service* é o comunicador entre *frontend* e *backend*.

É importante destacar a existência de *Web Service* voltados para os SIG seguindo definições de padrões abertos, estes por sua vez, normatizados pela OGC (Open Geospatial Consortium). Os principais são:

- ❖ *Web Map Service* (WMS) - "padrão que permite aos usuários acessar remotamente imagens de mapas georreferenciados por meio de solicitações de protocolo de transferência de hipertexto (HTTP)" (Jolaiya, 2020). Uma solicitação WMS define a (s) camada (s) geográfica (s) e a área de interesse para ser processado. A resposta à solicitação é um ou mais mapas (retornadas como JPEG, PNG etc.) que podem ser exibidas em um navegador (OGC, s.d.). Um WMS, sempre que requisitado, devolverá a informação em formato de imagem, possibilitando um maior desempenho na transmissão da informação, não consumindo muita memória.
- ❖ *Web Feature Service* (WFS) - permite a troca de dados geográficos em formato vetorial na Web. Os usuários podem criar, excluir, atualizar ou bloquear uma instância do recurso, (Jolaiya, 2020). Em vez de compartilhar informações geográficas no nível do arquivo usando *File Transfer Protocol* (FTP), por exemplo, o WFS oferece acesso direto e refinado às informações geográficas no nível do recurso e da propriedade do recurso (OGC, s.d).
- ❖ *Web Coverage Service* (WCS) – “funciona sobre o protocolo HTTP, definindo como obter dados e metadados usando as solicitações disponíveis nesse protocolo. Na prática,

permite que mapas em formato *raster* sejam obtidos de um navegador da web ou de qualquer outro programa que use o protocolo” (ISRIC, 2021).

Portanto, para finalizar, o sistema Web é composto pelo *frontend* (parte que o utilizador navega e interage com o sistema) e pelo *backend* (a “caixa negra”, que gere todo conteúdo da aplicação Web) sendo a comunicação entre eles feita através do *Web Service* e, os métodos HTTP responsáveis pelo transporte da informação. E para completar, podendo ser utilizada ou não, a base de dados para armazenamento de toda informação.

2.1.3 Base de dados relacional

Nesta área dos conceitos e definições é importante mencionar o que é uma base de dados relacional, para o que serve e sua linguagem de programação. Assim, segundo a instituição Oracle Brasil, uma base de dados relacional é:

“um tipo de banco de dados que armazena e fornece acesso a pontos de dados relacionados entre si. Bancos de dados relacionais são baseados no modelo relacional, uma maneira intuitiva e direta de representar dados em tabelas. Em um banco de dados relacional, cada linha na tabela é um registro com uma ID exclusiva chamada chave. As colunas da tabela contêm atributos dos dados e cada registro geralmente tem um valor para cada atributo, facilitando o estabelecimento das relações entre os pontos de dados.” (ORACLE, 2022).

“Estruturar dados desta forma torna o acesso eficiente e flexível” (AZURE, s.d), por meio da linguagem de programação SQL (*Structured Query Language*) é possível pedir os dados armazenados, armazenar novas informações, atualizar algum dado existente ou apagá-los. De fato, “para ser considerado uma base de dados, o local onde os dados são armazenados deve conter não apenas os dados, mas também informações sobre as relações entre esses dados” (Harrington, 2016).

Deste modo, este tipo de base de dados, permite criar relacionamentos entre as tabelas, através das chaves primárias e estrangeiras. Uma chave primária corresponde a uma ou várias colunas/atributos que não possuem valores duplicados dentro de uma tabela. Uma chave estrangeira corresponde a uma ou várias colunas em que os valores estejam identificados necessariamente como chave primária de outra tabela (Bonfioli, 2006, p.6), portanto, seguindo esta conceituação, a chave estrangeira é o mecanismo que define os relacionamentos em uma base de dados relacional (Date, 2000 citado em Bonfioli, 2006, p.6).

Existem ainda a “Base de Dados Geográfica (BDG) que, consiste numa coleção de dados geográficos de diversos tipos armazenados em sistemas de arquivos comuns. Numa BDG é possível, além de armazenar, fazer inquirições e manipulações sobre os dados” (Cosme, 2012, p. 238). Reunindo condições para a implementação de modelos relacionais entre os diferentes dados, tratando-se de “uma estrutura de dados especializada no armazenamento de dados geográficos, dos seus atributos e relacionamentos” (Cosme, 2012, p.12).

2.1.4 WebSIG

Com o avanço tecnológico e a necessidade de disseminar a informação, sobretudo relacionada ao espaço, tornou-se inevitável o desenvolvimento de aplicações Web com os SIG para publicar dados geoespaciais através da Internet e fornecer aos utilizadores funções de navegação, consulta, análise, inquirições e funções SIG (Li, 2020, p.120).

O WebSIG tem características especiais que o difere dos tradicionais programas SIG de *desktop* assim, de acordo com Liu N (citado em Li, 2020, p.120), os principais aspectos do WebSIG são:

- ❖ Um sistema cliente/servidor baseado em rede, enquanto o SIG tradicional é principalmente um sistema autónomo independente;
- ❖ Um sistema distribuído, onde usuários e servidores podem ser distribuídos em diferentes locais e em diferentes plataformas de computador;
- ❖ Usa a Internet para trocar informações entre o cliente e o servidor, o que significa que a transmissão de informações é global;
- ❖ Não tem restrições em nenhum computador ou sistema operacional. Os usuários podem acessar e usar o WebSIG desde que tenham acesso à Internet.

Existem várias maneiras de construir um WebSIG, destacando o mais usual que é o desenvolvimento baseando nas tecnologias do *backend* e do *frontend* (Li, 2020, p.121).

Quando o *backend* recebe um pedido do *frontend*, através do navegador, ele chama as bibliotecas geoespaciais, que por sua vez acessam os dados geográficos armazenados na base de dados, executa as funções SIG e retorna o resultado da execução para o *frontend*, que apresenta a informação estática ou não no navegador. Assim, tanto os dados geográficos como as funções de geoprocessamento ficam alocadas no *backend*. O *frontend* fica apenas responsável por enviar os

pedidos ao *backend* e apresentar ao utilizador o resultado correspondente ao pedido vindo do *backend* (Li, 2020, p.121).

Deste modo, pode ser afirmar que um WebSIG é um conjunto de protocolos, ferramentas, serviços Web e as linguagens de programação, desde o *backend* até o *frontend*, inserindo neste esquema: as funções dos Sistemas de Informação Geográfica através das bibliotecas geoespaciais (GDAL/OGR, Geopandas, GeoDjango etc.) e outras formas, para inserção e manipulação de dados geográficos; os servidores de mapa responsáveis pela criação dos serviços do *backend* e as bibliotecas em JavaScript para a visualização de mapa no *frontend*. Permitindo portanto, a interação com o usuário através de pedidos de informação dos dados geoespaciais e/ou alfanuméricos, importação e exportação de dados e a criação de mapas. Destacando também suas principais vantagens, que são: alcance global, maior número de usuários, baixo custo, fácil utilização para o usuário e fácil atualização/manutenção (Fu, 2018).

2.2 Estado da arte

Projetos relacionados com a criação de WebSIG no contexto académico e que foram dados como exemplos outrora, serviram como uma pequena amostra para o desenvolvimento deste, por não estarem ativos em rede, não serão citados. Contudo, outros trabalhos foram pesquisados fora do contexto académico (Tabela 5) e, estimularam a busca por conhecimentos para a elaboração deste projeto e realçando o quanto os SIG abrangem as mais diversas áreas de estudo.

Tabela 5: Alguns exemplos de WebSIG e suas principais funcionalidades.

URL	INSTITUIÇÃO	REGIÃO	PRINCIPAIS TEMAS
https://www.data.rio/apps/sig-floresta/explore	Prefeitura do Rio de Janeiro	Rio de Janeiro	Território e Meio Ambiente; Uso do Solo; População; Economia; Educação; Saúde; Cultura; Turismo; Esporte e Lazer; Infraestrutura e saneamento; Transporte; Assistência Social; Habitação e Urbanismo; Segurança Pública; Administração Pública
https://snic.dgterritorio.gov.pt/visualizadorCadastro	Direcção-Geral do Território	Portugal	Sistema Nacional de Informação Cadastral.
http://www.pnrc.pt/index.php/geovisualizador/	Plataforma Nacional para a Redução do Risco de Catástrofes.	Portugal	Risco de: Acidentes Aéreos; Acidentes Ferroviários; Acidentes Fluviais e Marítimos; Acidentes com Matérias Perigosas em Ferrovia; Acidentes com Matérias Perigosas em Instalação Fixa; Risco de Acidentes com Matérias Perigosas em Rodovia; Risco de Acidentes Rodoviários; Cheias Fluviais T100; Colapso de Edifícios com Elevada Concentração Populacional; Colapso de Pontes e Túneis; Emergências Radiológicas; Erosão Costeira; Galgamentos Costeiros; Incêndios Rurais; Incêndio em Centros Históricos; de Incêndios Urbanos; Deslizamentos de Massa; Nevões etc.
https://websig.cm-lisboa.pt/MuniSIG/visualizador/	Câmara Municipal de Lisboa	Lisboa	Ambiente; Cartografia; Edificado; Equipamentos; Espaço Público; Gestão Urbanística; Habitação etc.
https://www.london.gov.uk/what-we-do/research-and-analysis/webmaps-and-data-services	Prefeitura de Londres	Londres	Qualidade do ar; Infraestrutura para veículos elétricos; Serviços para cidadãos europeus; Cobertura do Solo; Vias de Circulação; Escolas; Saúde; Energia; Árvores frutíferas catalogadas; etc.
https://healthyregionseurope.uc.pt/#/view/map/model?y=2014	Centro de Estudos de Geografia e Ordenamento do Território – CEGOT Univ. de Coimbra (UC)	Europa	O principal objetivo é fornecer um informações sobre a saúde da população europeia em múltiplas dimensões e nível geográfico.
https://worldview.earthdata.nasa.gov/	National Aeronautics and Space Administration (NASA)	Global	Detecção remota para identificação de fogos; Luzes noturnas; Tempestades de areias; Ciclones tropicais; Erupções vulcânicas; Furacões; entre outros aspectos ligados aos fenômenos naturais.

Tabela 5: Alguns exemplos de WebSIG e suas principais funcionalidades. (Continuação)

URL	INSTITUIÇÃO	REGIÃO	PRINCIPAIS TEMAS
https://geocatalogo.icnf.pt/websig/	Instituto da Conservação da Natureza e das Florestas (ICNF)	Portugal	Organização territorial; Fauna, flora e habitats; Florestas; Caça e pesca; Riscos e ameaças; Turismo e lazer na natureza.
https://imexmap.mapidea.com/	Balança Comercial Portuguesa.	Portugal	Apresenta os valores de importação e exportação, desde 2012 a 2022, para Portugal e para os Concelhos portugueses.
https://idecentro.ccdrc.pt/VIG	Infraestrutura De Dados Espaciais Da Região Centro De Portugal.	Região Centro de Portugal	Cartografia: Limites Administrativos, Geologia, Geomorfologia e Paisagem, Militares, Uso e Ocupação do Solo; Território, Economia e Desenvolvimento: Economia e Inovação, Infraestruturas e Equipamentos, Turismo e Lazer, Cultura e Património; Ambiente: Atlas do Ambiente, Recursos Hídricos, Infraestruturas, Serviços e Equipamentos, Conservação e Proteção da Natureza, Riscos e catástrofes etc.
http://saudemunicipio.uc.pt/app/#/map/ines?y=2011&i=ines	Índice da Saúde da População	Portugal	Segue as determinantes: Económicos E Sociais, Ambiente Físico, Estilos De Vida, Cuidados De Saúde; para obter os Resultados: Morbilidade e Mortalidade, para cada Concelho de Portugal, para medir o índice da saúde da população.
https://myocean.marine.copernicus.eu/	Copernicus	Global	Indica vários parâmetros relacionados aos oceanos como: a temperatura da água, salinidade da água, espessura do gelo marinho, concentração de clorofila na água do mar etc.
http://sigmealhada.cm-mealhada.pt/	Câmara Municipal da Mealhada	Mealhada	Cartografia de base; Planos municipais de ordenamento do território; Servidões e restrições de utilidade pública; Áreas de reabilitação urbana; Serviços públicos e equipamentos coletivos; Informação turística; Floresta, proteção civil e ambiente.
http://epic-webgis-portugal.isa.ulisboa.pt/	Ecological Planning, Investigation And Cartography – EPIC WebGIS Portugal	Portugal	Informação administrativa, relevo, morfologia do terreno, corpos d'água, solo e subsolo, vegetação, conservação da natureza, litoral, estrutura ecológica nacional, aptidão agroecológica às culturas agrícolas, aptidão bioclimática às espécies arbóreas, vias romanas, área edificada e infraestruturas.

Dentre todos os citados anteriormente, destacam-se o Euro Healthy (Figura 2) desenvolvido pelo CEGOT, o Georubus produzido pelo ICNF e o da Balança Comercial Portuguesa, que por suas funcionalidades, remetem aquilo que se almeja apresentar através do WebSIG deste projeto, designado pelo nome Christian World Mapping (CWM).

Os *dropdowns* da página do Euro Healthy, por exemplo, inspiram a construir um semelhante com as informações referentes aos anos, territórios e aos indicadores, dando as opções de escolha para o utilizador. Este WebSIG, assim como se espera no CWM, relaciona as informações alfanuméricas com a informação geográfica.

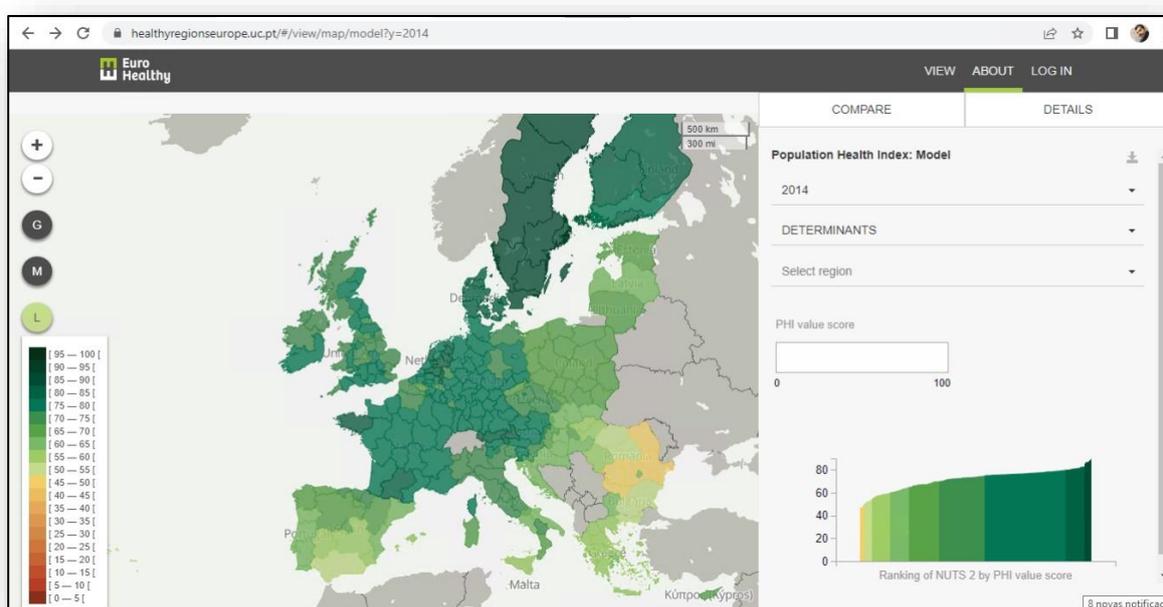


Figura 2: Euro Healthy, exemplo de WebSIG.

A localização do mapa à esquerda, fugindo do tradicional que é à direita, a princípio proporciona visualmente uma melhor apresentação da informação geoespacial e deixando o espaço do lado direito para as informações alfanuméricas.

No WebSIG da Balança Comercial Portuguesa (Figura 3), outra informação apresentada além do mapa à esquerda e que encoraja uma implementação semelhante no CWM, é o gráfico temporal que surge conforme o utilizador altera o Concelho a ser consultado, aumentando à informação transmitida. No caso do CWM, o gráfico será para a representação dos diferentes tipos de indicadores.

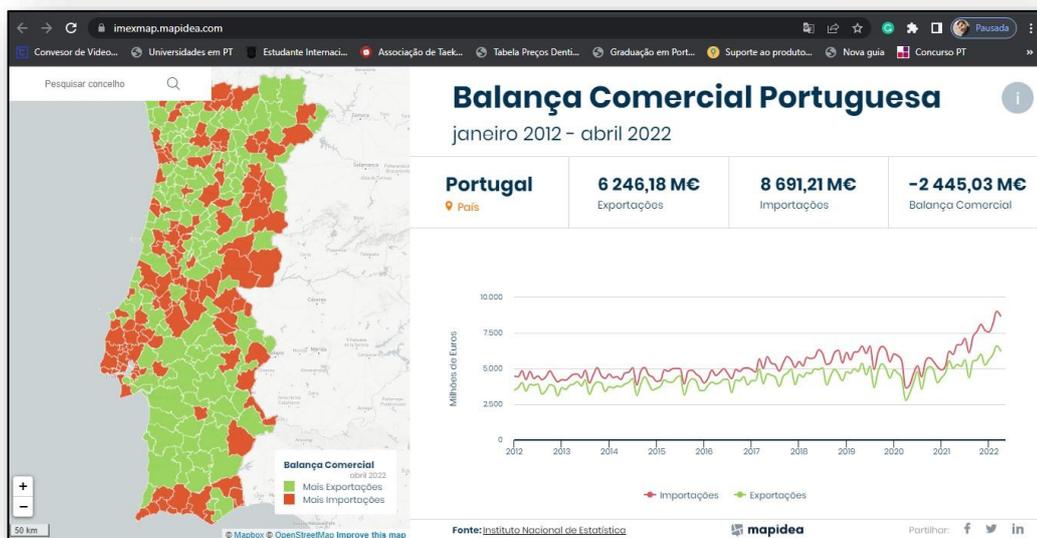


Figura 3: Balança Comercial Portuguesa, exemplo de WebSIG.

Observou-se o mapa de base apresentado no Georubus do ICNF (Figura 4), especificadamente o de estilo *National Geographic* produzido pela ESRI, e percebeu-se que talvez seja uma opção a ser implementada no CWM, proporcionando ao utilizador um mapa de base com mais cores e por não ser tão usual.

Após analisar todos os WebSIG citados na Tabela 5, foi possível extrair alguns bons exemplos para encorajar no desenvolvimento do CWM.

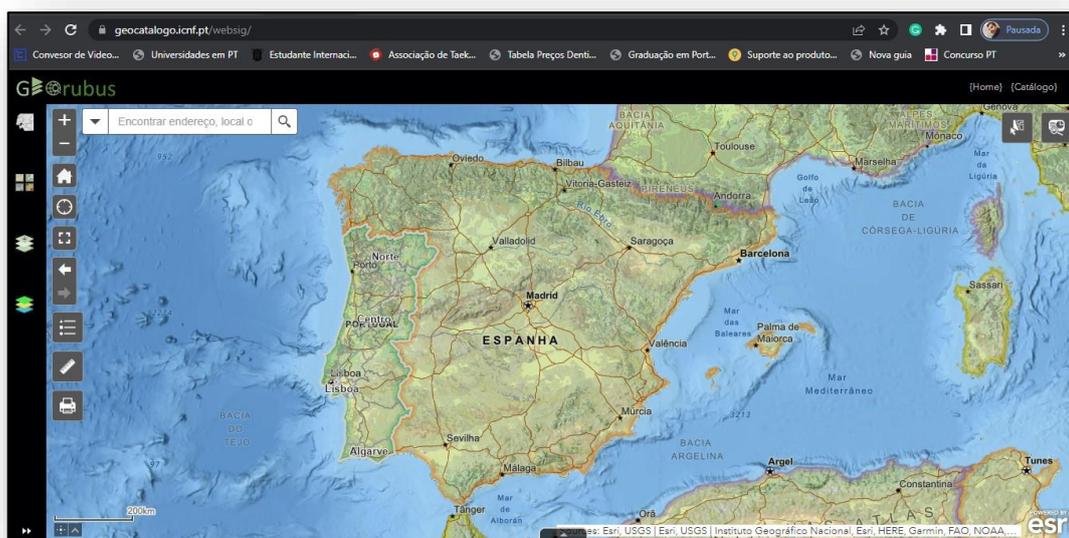


Figura 4: Georubus (ICNF), exemplo de WebSIG

2.3 A fonte e os dados sobre o crescimento do cristianismo

Para a criação deste projeto era preciso uma fonte de dados fiável, após algumas pesquisas na Web encontrou-se a instituição WCD, pertencente a Brill, editora académica fundada em 1683. A WCD é constituída principalmente pelo Doutor Todd M. Johnson, Professor de Missão e Cristianismo Global, Co-Diretor do Centro para o Estudo do Cristianismo Global no Seminário Teológico Gordon-Conwell, pesquisador visitante do Instituto de Cultura, Religião e Assuntos Mundiais da Universidade de Boston e, lidera o projeto de pesquisa sobre demografia religiosa internacional. Faz parte também da WCD a Doutora Gina A. Zurlo, demógrafa da religião e estuda todas as religiões em todos os países do mundo. É codiretora do Centro para o Estudo do Cristianismo Global do Seminário Teológico Gordon-Conwell, bem como pesquisadora visitante do Instituto de Cultura, Religião e Assuntos Mundiais da Universidade de Boston.

Com relação as metodologias utilizadas pela WCD para se chegar aos resultados dos números de cristãos em cada período, eles utilizam:

- ❖ Censos que contenham perguntas sobre religião e os que contenham perguntas sobre etnia e língua (por considerarem a influência de uma localidade e língua para a escolha de uma religião);
- ❖ Pesquisas e inquéritos nas mais distintas instituições globais, continentais, nacionais ou regionais (como a ONU, UNICEF etc.);
- ❖ Monografias académicas;
- ❖ Estatísticas sobre religião em anuários e manuais e relatórios estatísticos governamentais;
- ❖ Questionários e relatórios feitos por colaboradores;
- ❖ Pesquisas de campo e entrevistas;
- ❖ Comparação de informações com instituições de ensino que tenham dados sobre a religião, eliminando as discrepâncias entre dados;
- ❖ Documentos não publicados (documentos coletados no campo e incluem relatórios, memorandos, fotocópias, fotografias, mapas, resumos estatísticos e documentos históricos);
- ❖ Enciclopédias, dicionários e diretórios de religião;
- ❖ Dissertações e teses sobre as religiões (Johnson & Grim, 2013).

Por meio de contacto via e-mail entre o autor e a Dra. Gina no dia 15 de março de 2021, solicitando permissões para aceder à base de dados, justificando o objetivo do trabalho académico e informando que os dados pertencentes à WCD seriam de suma importância para a criação do projeto. Foi disponibilizado o acesso à base de dados durante 30 dias, período suficiente para extração dos dados necessários.

Voltando a questão inicial em como será a espacialização da expansão do cristianismo durante os anos, a dificuldade em definir os limites territoriais se dá pelos inúmeros eventos ocorridos durante a história da população mundial, as invasões ou conquistas de povos sobre outros povos, impérios que surgiram e que desapareceram, as colonizações etc. os limites geográficos foram se alterando. Mesmo no século XX após a Segunda Guerra Mundial, muitos países tiveram seus limites territoriais modificados, fato que ocorreu também com o fim da União Soviética. Hoje se houver alguma transformação dos limites territoriais é mais fácil obter dados para cartografar estas alterações, mas, se tratando de mudanças que ocorreram há séculos passados seria um trabalho árduo para cartografar. Então, uma solução para este desafio foi ancorar este projeto na fonte de dados, que por sua vez definiu os limites territoriais em quatro escalas para dimensionar a expansão do cristianismo: “Mundo”, “Continentes”, “Regiões do mundo” e “Países” (Tabela 6), ou seja, com base no que foi estabelecido pela WCD para a produção dos dados alfanuméricos para cada tipo de território é que definiram-se os limites geográficos dos dados geoespaciais deste projeto.

Tabela 6: Exemplo de separação dos dados por territórios segundo a WCD.

GLOBAL	CONTINENTE	REGIÃO DO MUNDO	PAÍSES
Mundo	Europa	Norte da Europa	Ilhas do Canal
			Dinamarca
			Estônia
			Ilhas Faroé
			Finlândia
			Islândia
			Irlanda
			Ilha de Man
			Letônia
			Lituânia
			Noruega
			Suécia
Reino Unido			

Durante o acesso a plataforma, o que se extraiu foram ficheiros em formato .xlsx (Excel) (Figura 5) com as informações distribuídas por territórios, por anos e pelos seguintes indicadores: população do território; população cristã; percentagem do número de cristãos com relação a população; destes cristãos o número e percentagem dos que eram/são/serão católicos, ortodoxos, protestantes, independentes, não afiliados e duplamente afiliados; e pessoas não evangelizadas.

ID	UN Region	Year	Population	Christian Pop	Christian Pct	Orthodox Pop	Orthodox Pct	Catholic Pct
38	37 Eastern Africa		33	1,20	0,00	0,00%	0,00	0,00%
39	38 Eastern Africa		100	1,40	0,00	0,00%	0,00	0,00%
40	39 Eastern Africa		200	1,70	0,00	0,00%	0,00	0,00%
41	40 Eastern Africa		300	2,20	0,01	0,45%	0,01	100,00%
42	41 Eastern Africa		400	2,50	0,30	12,00%	0,30	100,00%
43	42 Eastern Africa		500	2,80	0,50	17,86%	0,50	100,00%
44	43 Eastern Africa		600	3,10	0,50	16,13%	0,50	100,00%
45	44 Eastern Africa		700	3,50	0,60	17,14%	0,60	100,00%
46	45 Eastern Africa		800	3,90	0,70	17,95%	0,70	100,00%
47	46 Eastern Africa		900	4,50	0,80	17,78%	0,80	100,00%
48	47 Eastern Africa		1000	5,40	0,90	16,67%	0,90	100,00%
49	48 Eastern Africa		1100	6,70	1,00	14,93%	1,00	100,00%
50	49 Eastern Africa		1200	8,10	1,20	14,81%	1,20	100,00%
51	50 Eastern Africa		1300	8,50	1,40	16,47%	1,40	100,00%
52	51 Eastern Africa		1350	9,70	1,60	16,49%	1,60	100,00%
53	52 Eastern Africa		1400	10,70	1,70	15,89%	1,70	100,00%
54	53 Eastern Africa		1500	11,50	1,80	15,65%	1,80	100,00%
55	54 Eastern Africa		1550	12,30	1,90	15,45%	1,90	100,00%
56	55 Eastern Africa		1600	13,20	2,00	15,15%	2,00	100,00%
57	56 Eastern Africa		1650	14,10	2,20	15,60%	2,19	99,55%

Figura 5: Exemplo de como vieram os dados da WCD.

É de ressaltar que nem todos os tipos de territórios tem os mesmos anos e nem os mesmos indicadores, conforme é apresentado na Tabela 7.

Quanto ao tipo de território, “Regiões do mundo”, a sua definição deu-se seguindo as informações contidas na folha de Excel (Figura 6), contendo o atributo da tabela “Country Name” e o atributo “UN-Region”, então o que se fez foi filtrar cada região do mundo para descobrir quais os países que nela pertencem, para que posteriormente através de um *software* SIG pudesse ser feito a criação deste tipo de território.

1	A	B	C	D	E	F	G	H	I
1	Country Name	UN Region	Year	Christians	Christian%	Catholics	Independents	Orthodox	Protestants
2	Afghanistan	South Asia	2020	7.540	0,02%	200	4.200	90	2.550
3	Albania	Southern Europe	2020	1.081.740	36,77%	515.000	28.000	530.000	13.000
4	Algeria	Northern Africa	2020	129.350	0,30%	6.500	110.000	1.300	11.400
5	American Samoa	Polynesia	2020	54.429	97,54%	16.000	21.500	0	41.960
6	Andorra	Southern Europe	2020	69.454	89,98%	65.291	600	0	140
7	Angola	Middle Africa	2020	30.591.841	93,19%	17.700.000	2.055.000	0	9.000.000
8	Anguilla	Caribbean	2020	13.588	88,91%	720	1.360	0	10.800
9	Antigua & Barbuda	Caribbean	2020	97.950	93,19%	8.600	3.400	0	65.200
10	Argentina	South America	2020	40.430.679	88,84%	35.500.000	2.604.000	170.000	2.465.100
11	Armenia	Western Asia	2020	2.774.179	94,40%	250.000	66.000	2.477.000	70.000
12	Aruba	Caribbean	2020	101.728	95,57%	81.360	4.500	0	11.140
13	Australia	Australia-New Zealand	2020	13.744.477	54,12%	5.900.000	800.000	1.100.000	5.860.000
14	Austria	Western Europe	2020	6.278.400	71,49%	5.327.000	83.000	205.000	363.300
15	Azerbaijan	Western Asia	2020	245.950	2,44%	650	5.000	220.000	19.450
16	Bahamas	Caribbean	2020	378.299	92,98%	53.500	30.000	500	295.500
17	Bahrain	Western Asia	2020	195.200	11,50%	160.000	15.200	7.000	13.000
18	Bangladesh	South Asia	2020	903.322	0,53%	420.000	310.000	150	360.000
19	Barbados	Caribbean	2020	272.283	94,68%	11.000	22.200	500	174.000
20	Belarus	Eastern Europe	2020	7.409.060	78,69%	1.035.000	37.000	5.675.000	250.000
21	Belgium	Western Europe	2020	7.285.371	62,70%	7.000.000	103.000	65.500	166.000
22	Belize	Central America	2020	368.527	92,59%	284.000	25.000	0	124.000
23	Benin	Western Africa	2020	5.631.494	46,45%	2.970.000	1.470.000	0	1.250.000
24	Bermuda	Northern America	2020	52.439	86,48%	9.300	9.100	0	34.300

Figura 6: Exemplo da associação dos países com as “Regiões do mundo”.

Tabela 7: Distribuição dos dados por territórios, anos e indicadores.

Territórios	Anos	Indicadores
“Mundial” (1)	33, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1350, 1400, 1500, 1550, 1600, 1650, 1700, 1750, 1800, 1850, 1900, 1910, 1950, 1970, 2000, 2005, 2010, 2015, 2020, 2025, 2030 e 2050.	População, Cristãos, Percentagem de Cristãos, número e percentagem dos cristãos católicos, ortodoxos, protestantes, independentes, não afiliado e pessoas não evangelizadas.
“Continentes” (5)	33, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1350, 1400, 1500, 1550, 1600, 1650, 1700, 1750, 1800, 1850, 1900, 1910, 1950, 1970, 2000, 2005, 2010, 2015, 2020, 2025, 2030 e 2050.	População, Cristãos, Percentagem de Cristãos, número e percentagem dos cristãos católicos, ortodoxos, protestantes, independentes, não afiliado e pessoas não evangelizadas.
“Regiões do mundo” (22)	33, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1350, 1400, 1500, 1550, 1600, 1650, 1700, 1750, 1800, 1850, 1900, 1910, 1950, 1970, 2000, 2005, 2010, 2015, 2020, 2025, 2030 e 2050.	População, Cristãos, Percentagem de Cristãos, número e percentagem dos cristãos católicos, ortodoxos, protestantes, independentes, não afiliado e pessoas não evangelizadas.
“Países” (234)	1900, 1910, 1950, 1970, 2000, 2005, 2010, 2015, 2020, 2025, 2030 e 2050.	Cristãos, número e percentagem dos cristãos católicos, ortodoxos, protestantes, independentes e não afiliado.

Seguindo o exemplo da Figura 4, encontram-se no Anexo B as Tabelas 11, 12, 13, 14 e 15 que apresentam todos os países da atualidade inseridos em cada uma das seguintes “Regiões do mundo”:

- África Ocidental, África Central, África do Norte, África Austral e África Oriental; América do Sul, América do Norte, América Central, Caribe; Ásia Central, Ásia Oriental, Ásia Ocidental, Sudeste asiático e Sul da Ásia; Europa Ocidental, Europa Oriental, Norte da Europa e Sul da Europa; Austrália-Nova-Zelândia; Melanésia; Micronésia; Polinésia.

Por fim, para dar seguimento ao desenvolvimento do WebSIG, extraiu-se da EUROSTATIC (*European Statistics*) os dados geoespaciais em formato ESRI Shapefile (SHP) com as informações geográficas dos países da atualidade, na projeção cilíndrica de Mercator e com o sistema de coordenadas *World Geodetic System 1984* (WGS84). Posteriormente manipulado para criar um ficheiro SHP nas escalas: “Continentes” e “Regiões do mundo”.

3. Arquitetura do sistema, casos de uso e respectivas tecnologias

Nesse capítulo o que se pretende é expor de forma clara como será desenvolvido o WebSIG, desde os requisitos funcionais da aplicação, a estrutura do sistema, como cada parte se comunica, as tecnologias a serem utilizadas, e por fim, falar sobre o domínio da página, o idioma utilizado e um esboço de como será o *layout* da página de boas-vindas e da página principal.

3.1 Requisitos funcionais

Nesta secção serão apresentados os requisitos funcionais, indicando as partes essenciais e obrigatórias do WebSIG. Assim, a aplicação estará dividida nas seguintes partes: a gestão do utilizador, a obtenção das informações, a da gestão das informações pelo administrador da página e a gestão dos dados no GeoServer.

Serão dois tipos de utilizadores do sistema, também chamados de atores, o *Client User* e o *Admin User*, o primeiro tem acesso a aplicação apenas para fazer pedidos utilizando o método http POST quando fizer o login. Outro pedido que estará disponível para o *Client User* será o GET, para obter as informações vindas da base de dados referentes aos tipos de territórios (“Mundo”, “Continentes”, “Regiões do mundo” ou “Países”), aos anos, aos indicadores e aos territórios disponíveis para cada tipo de território. Observe as Tabelas 8 e 9, que descrevem os requisitos funcionais para este utilizador.

Entretanto, ressalta-se que este utilizador só acessará a página principal do sistema WebSIG quando tiver as credenciais (*username* e *login*) do *Admin User*. O fato pelo qual o *Client User* precisa destas credenciais para aceder a aplicação, se dar pelo motivo que os dados extraídos da WCD, encontram-se fechados e por isso, será preciso formalizar um pedido de autorização a ela para que esses dados possam ser abertos a todos os utilizadores.

Quanto ao utilizador *Admin User*, ele poderá aplicar as funções CRUD para: todos os *endpoints* da aplicação, que estarão relacionados à disponibilização da informação alfanumérica e geográfica no WebSIG; fará a gestão dos utilizadores, dando-lhes as credenciais de acesso e gerir as permissões do utilizador; fará a gestão dos dados no Geoserver (pedir, criar, atualizar e apagar *workspaces*, *stores*, *layers* e *styles*); e quando necessário fará as atualizações do sistema. Da Tabela 16 até a Tabela 26 que estão no Anexo C, é possível verificar os requisitos funcionais deste utilizador.

O diagrama da Figura 7, ilustra como cada ator fará a ligação com o requisito funcional (RF), este contendo uma identificação, que posteriormente pode ser encontrado numa das tabelas dos requisitos funcionais.

Quando houver uma ligação entre o ator “*Admin User*” e uma caixa de gestão, essa ligação utilizará o “*<<extend>>*”, indicando que qualquer mudança que ocorra, esta será para um determinado caso, ou para um determinado registo na tabela da base de dados.

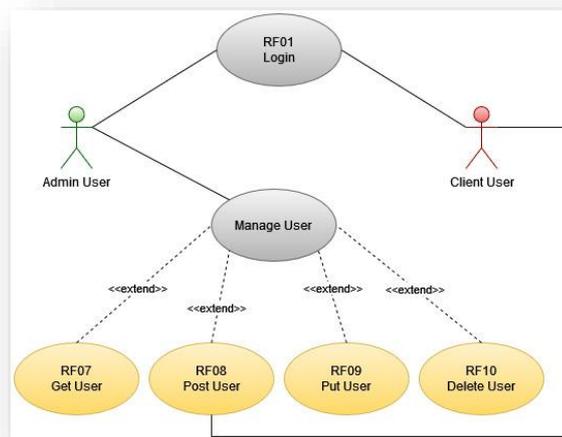


Figura 7: Utilizador e gestão do utilizador.

Já o diagrama da Figura 8 surge para explicar e representar a interação dos utilizadores com o sistema, quando estiverem na página principal. Os requisitos funcionais 02, 03, 04, 05 e 06 devolvem uma listagem com os valores de cada chave, para que seja possível as seleções pelo utilizador.

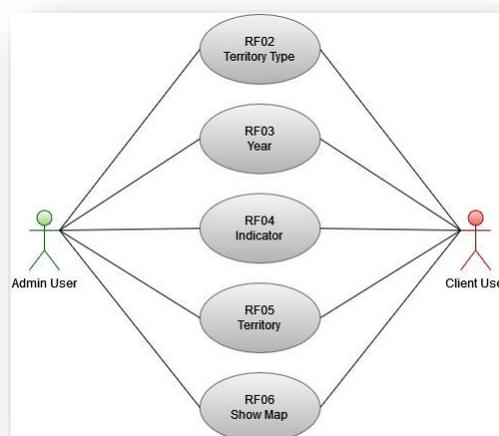


Figura 8: RF's que interagem com o utilizador.

Tabela 8: Requisitos funcionais da aplicação, página login e página principal.

RF	ATOR	NOME	MÉTODO HTTP	ONDE	CONDIÇÃO	FLUXO PRINCIPAL	FLUXO SECUNDÁRIO
01	<i>Client User;</i> <i>Admin User</i>	<i>Login</i>	POST	Página Login	O utilizador tem que ter as credenciais de acesso;	1 – O utilizador preenche os campos <i>username</i> e <i>password</i> ; 2 – Clica no botão <i>Login</i> ; 3 – O sistema confirmará a autenticação e a permissão; 4 – O utilizador é redirecionado para a página principal.	1 – Caso insira <i>username</i> ou <i>password</i> errado, recebe uma mensagem de erro; 2 – Caso não tenha as credenciais de acesso, o sistema enviará uma mensagem a dizer que precisa delas para aceder a página principal.
02	<i>Client User;</i> <i>Admin User</i>	<i>Choose a Territory Type</i>	GET	Página principal	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador escolhe obrigatoriamente no <i>dropdown</i> uma opção referente ao tipo de território; 2 – Clica sobre o nome escolhido; 3 – Próximo <i>dropdown</i> é disponibilizado para o utilizador.	1 – Caso o utilizador não seleccione o tipo de território o sistema não permitirá aceder ao outro <i>dropdown</i> com os dados referente aos anos.
03	<i>Client User;</i> <i>Admin User</i>	<i>Choose a Year</i>	GET	Página principal	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador escolhe obrigatoriamente no <i>dropdown</i> uma opção referente ao ano; 2 – Clica sobre o nome escolhido; 3 – Próximo <i>dropdown</i> é disponibilizado para o utilizador.	1 – Caso o utilizador não seleccione um determinado ano o sistema não permitirá aceder ao outro <i>dropdown</i> com os dados referente aos indicadores.

Tabela 9: Requisitos funcionais da aplicação, página principal.

RF	ATOR	NOME	MÉTODO HTTP	ONDE	CONDIÇÃO	FLUXO PRINCIPAL	FLUXO SECUNDÁRIO
04	<i>Client User;</i> <i>Admin User</i>	<i>Choose a Indicator</i>	GET	Página principal	Utilizador com as credenciais e permissões autenticadas pelo sistema.	<p>1 – Utilizador escolhe obrigatoriamente no <i>dropdown</i> uma opção referente ao indicador;</p> <p>2 – Clica sobre o nome escolhido;</p> <p>3 – Próximo <i>dropdown</i> é disponibilizado para o utilizador.</p> <p>4 – O mapa é apresentado ao utilizador segundo as opções escolhidas por ele.</p>	<p>1 – Caso o utilizador não seleccione um determinado indicador o sistema não permitirá aceder ao outro <i>dropdown</i> com os dados referente aos territórios.</p> <p>2 – Caso o utilizador não seleccione um determinado indicador o sistema não apresentará o mapa de acordo com as opções escolhidas.</p>
05	<i>Client User;</i> <i>Admin User</i>	<i>Choose a Territory</i>	GET	Página principal	Utilizador com as credenciais e permissões autenticadas pelo sistema.	<p>1 – Utilizador escolhe opcionalmente no <i>dropdown</i> uma opção referente ao território;</p> <p>2 – Clica sobre o nome escolhido;</p> <p>3 – O sistema mostrará ao utilizador o valor numérico correspondente aquele território, de acordo com as opções, ano e indicador, escolhidas anteriormente.</p>	<p>1 – Caso o utilizador não seleccione um determinado território o sistema não devolverá a informação numérica sobre o território.</p>
06	<i>Client User;</i> <i>Admin User</i>	<i>Show Map</i>	GET	Página principal	Utilizador com as credenciais e permissões autenticadas pelo sistema.	<p>1 – Após a escolha das opções: tipo de território, ano e indicador, o sistema automaticamente apresentará o mapa, com o <i>style</i> segunda as classes pré-definidas, a legenda para identificar cada região.</p> <p>2 – Quando o utilizador altera qualquer uma das opções o sistema apaga o mapa que está na apresentação, o substitui pelo novo mapa, para não haver sobreposições.</p>	<p>1 – Caso o utilizador não preencha os campos com as opções, o sistema não devolve nenhum mapa.</p>

A gestão do tipo de território, gestão do ano, gestão do indicador, gestão do território, gestão da *layer* e a gestão da população (Figura 9), o ator *Admin User* poderá fazê-los através da página *Administrator* do próprio Django REST framework, nela a inserção de novos dados, a leitura, a alteração ou a eliminação de algum dado existente, será feita manualmente, preenchendo os campos requisitados. Quando houver um fluxo maior de informação, estes processos deverão ser automatizados por meio de *scripts* Python.

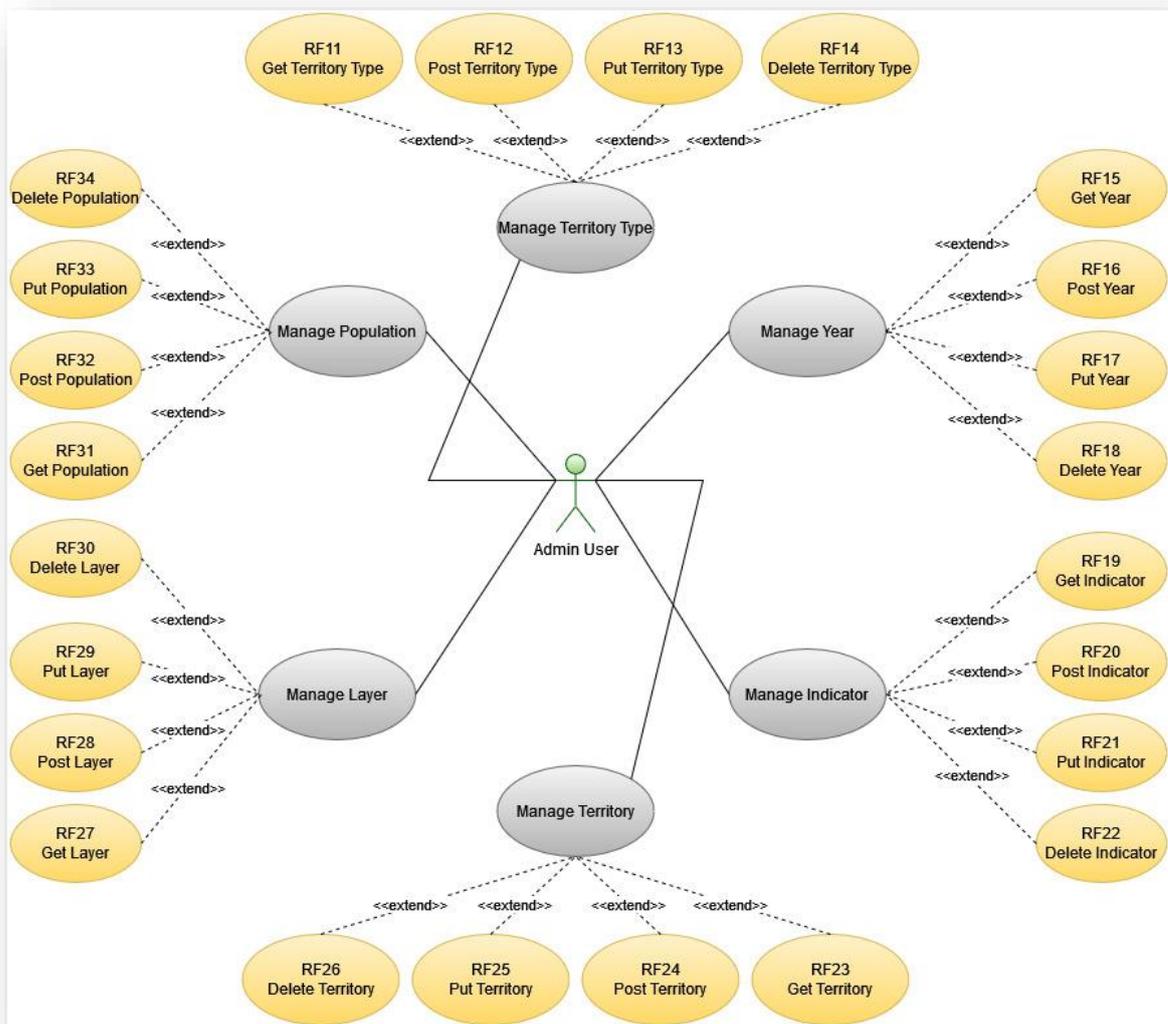


Figura 9: Gestão dos dados alfanuméricos e geoespaciais.

A gestão dos dados vinculados ao GeoServer (Figura 10), assim como a gestão dos dados mencionados anteriormente, os processos utilizando as funções CRUD para o *Workspace*, *Store*, *Layer* e *Style* serão automatizados, utilizando para o efeito a biblioteca Python “*geo.Geoserver*”.

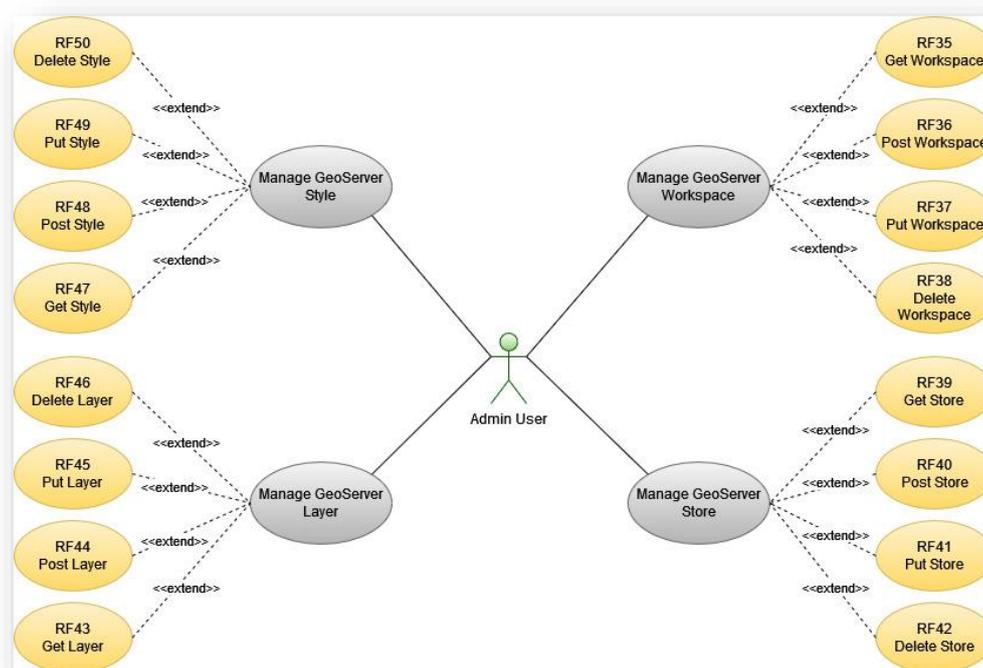


Figura 10: Gestão do GeoServer.

3.2 Tecnologias

Primeiramente destaca-se o sistema operacional no qual todo projeto se desenvolverá, o Linux Ubuntu 20.04 será instalado na máquina virtual VMware Workstation 16. Através de seu terminal de comando o Linux possibilita a instalação rápida, segura e eficaz de todos os programas necessários, sendo a melhor opção, comparando com o sistema operacional Windows, para avançar com projeto.

Para o desenvolvimento da aplicação através da edição de linhas de código o *software* a ser utilizado será o editor de texto Visual Studio Code, que permiti uma melhor estrutura e é intuitivo em cada tipo de linguagem de programação.

As próximas secções apresentarão as tecnologias que serão utilizadas para o desenvolvimento da base de dados relacional, do *backend*, do *frontend* e do servidor de mapas.

3.2.1 PostgreSQL - PostGIS

Por utilizar dados geoespaciais no projeto a base de dados relacional escolhida é a PostgreSQL, que utiliza a linguagem SQL, uma linguagem padrão para manipulação de tabelas e

registos numa base de dados relacional, sua extensão PostGIS permite o armazenamento, edição e aplicação de filtros e de operações topológicas a conjuntos de dados geoespaciais, viabilizando a relação com os dados alfanuméricos sobre o cristianismo. É um programa *open source*, que por sua vez não acarretará custos para o desenvolvimento do projeto e possibilitará a integração com os demais *software*. Outra possibilidade seria o sistema de gestão de base de dados, também relacional SQLite, que tem a extensão SpatiaLite para os dados geoespaciais, porém sua capacidade de armazenamento é inferior ao PostGIS, limitando o seu desempenho, ao mesmo tempo que não permite acessos simultâneos e/ou concorrenciais.

A estrutura da base de dados relacional, contará com as seguintes tabelas: “Territory Type”, onde serão armazenados os tipos de territórios; a tabela “Territory”, que armazenará os dados geoespaciais que representam os “Continentes”, as “Regiões do mundo” e os “Países”; a tabela “Year”, que armazenará os anos ; a tabela “Indicator”, armazenará todos os indicadores disponíveis; a tabela “Layer” armazenará o nome do *workspace*, do *store* e da *layer*, que posteriormente será utilizada para chamar os serviços de mapa; e por fim a tabela de relacionamento “Population”, onde os quatro primeiros atributos da tabela serão designados pelos identificadores das tabelas anteriores através das chaves estrangeiras. Contará também com o atributo “Values”, este receberá os valores com a quantidade de pessoas ou percentagem de acordo com cada indicador, num respectivo ano e um determinado território.

3.2.2 Django REST framework

Existem várias maneiras de se construir um *backend* de um sistema Web, porém para o desenvolvimento desta aplicação, optou-se pela utilização do Django REST framework (DRF), programa *open source*, que dispõe de componentes agrupadas no sistema que viabiliza a utilização dos métodos HTTP, construção da base de dados e a iteração com o sistema *frontend* e os serviços de mapas.

A linguagem de programação utilizada no Django REST é o Python, que é a mesma linguagem de programação usada nos Sistemas de Informação Geográfica, ou seja, já existe uma familiarização do autor com este tipo de linguagem, não necessitando então, despender tempo para aprender uma nova linguagem para a construção do *backend*, mas sim, uma aprimoração dos conhecimentos em Python.

Outra justificação para a escolha do DRF, é a integração do sistema com os SIG, seja para a criação da base de dados PostGIS, ou pela manipulação dos dados geográficos e alfanuméricos através das funções básicas CRUD (*Create, Read, Update and Delete*), existindo

portanto, um número considerável de bibliotecas geoespaciais Python que permitem fazer operações com os dados geoespaciais. Assim, usar Django permite integrar métodos existentes noutras bibliotecas para transformação de coordenadas, aplicação de filtros com base na localização geográfica etc. Isto não seria possível usando outros *frameworks* como Symphony ou Java Spring, por exemplo.

Quanto a esta manipulação inicial dos dados (inserção, leitura, atualização ou eliminação dos dados alfanuméricos e/ou geoespaciais junto à base de dados) ela será feita através do próprio DRF, na página “Administration” ou pela automatização de processos através de *scripts* em Python.

O DRF é constituído pelos principais módulos:

- ❖ “Models” – responsável pela criação de tabelas e seus atributos, através das classes em Python;
- ❖ “Serializers” – responsável pela serialização dos dados inseridos na base de dados e extração deles quando requisitados, utilizando o formato JSON;
- ❖ “Views” – responsável pelas funcionalidades, são os *endpoints* do sistema, onde se defini o que será devolvido para *frontend*;
- ❖ “URL” – responsável pelo mapeamento, são os *routers* do sistema, identifica cada funcionalidade para posteriormente serem requisitadas.

3.2.3 Angular

Para o desenvolvimento do *frontend* o *framework* a ser utilizado será o Angular 11.1.2, um programa *open source*, que utiliza linguagens de programação: HTML, para a construção das páginas Web, é o documento enviado do *backend* para o navegador (*browser*) quando requisitado; utilização do CSS (*Cascading Style Sheets*), para a formatação dos estilos da página, ou seja, a definição das cores, fontes, *layout* da página, *background*, etc.; e a linguagem TypeScript, que permite criar as funções de interação com o utilizador durante a navegação e, criação de funções que permitirá a comunicação com o DRF. A escolha justifica-se pela familiarização do autor com o *framework*.

Quanto a visualização dos mapas, será utilizado o Leaflet, uma biblioteca JavaScript, que permiti ser integrada junto ao Angular. Através dele é possível inserir mapas de base e visualizar os serviços WMS, que serão requisitados pelo DRF ao servidor de mapas quando pedido pelo utilizador. A utilização do Leaflet e não de outro, se dá por causa das funcionalidades do CWM

serem relativamente básicas. Deste modo, o Leaflet é suficiente por ser mais “leve”, assim, possibilitando um melhor resultado.

3.2.4 GeoServer

Para criação dos serviços de mapas, neste projeto será utilizado o GeoServer, programa *open source*, que possibilita a criação de *layers* e *styles* de mapas. Quando eles são solicitados pelo *frontend* o GeoServer devolve os serviços de mapas, no caso deste projeto os serviços serão os WMS em formato de imagem (png, jpeg etc.). Utiliza-se o GeoServer pelo fato de existir bibliotecas em Python permitindo à criação das *layers* através da automatização de processos.

O ambiente do GeoServer é constituído pelo *Workspace*, local onde é definido o espaço de trabalho; o *Store*, lugar para definir como o GeoServer buscará os dados geoespaciais à base de dados PostGIS. É neste lugar do sistema que se executa as *queries* (consultas à base de dados) que definirão os parâmetros utilizados para a criação de cada *layer*.; o ambiente *Layer*, distinguida por ser o espaço que defini como será a apresentação da informação geoespacial, se será um serviço WMS ou WFS, o sistema de coordenadas etc.; por fim, o espaço *Style*, destinado ao carregamento do ficheiro *sld* (*Styled Layer Descriptor*) que segue a estrutura XML, e a sua associação com a respectiva *layer*. Ressalta-se que estes ficheiros serão criados através da automatização de processos em Python, de acordo com o número de classes para cada tipo de território.

3.3 Estrutura do sistema

Após compreender cada função do sistema e as tecnologias a serem utilizadas, explicaremos nessa secção, como que a informação circulará entre as partes. Mesmo que seja de modo resumido, a intenção não é se detém muito nesta secção, porque, mais adiante na fase de implementação a abordagem sobre a estrutura do sistema será maior e mais detalhada.

Cada uma destas partes do sistema, estarão isoladas no servidor e identificadas pelo número do “*Port*”, a porta de acesso (Tabela 10). E a comunicação entre o DRF, o Angular e o GeoServer será através do protocolo HTTP, portanto, fazem e recebem pedidos uns dos outros.

Tabela 10: Localização de cada tecnologia no servidor.

TECNOLOGIA	NÚMERO DO PORT
ANGULAR	4300
DJANGO REST FRAMEWORK	8000
GEOSERVER	8080
POSTGRESQL POSTGIS	5432

Assim, seguindo o exemplo da Figura 11, através da interface gráfica do utilizador, o usuário do CWM fará o pedido à aplicação do cliente, no caso o Angular, que comunicará este pedido utilizando os métodos http ao Django REST framework. Após receber o pedido, se ele for relacionado aos dados alfanuméricos o DRF buscará o que foi solicitado diretamente à base de dados e devolverá a informação em JSON (*JavaScript Object Notation*), formato leve para troca de informações entre os sistemas.

Quando a informação solicitada for relacionada aos dados geoespaciais, o DRF, comunica o pedido ao GeoServer, sendo esta uma forma encontrada para isolar o sistema do GeoServer, ou seja, a aplicação do *frontend* não poderá fazer o pedido diretamente a ele, toda informação passará primeiro pelo DRF, prevenindo qualquer tipo de invasão. O GeoServer por sua vez, busca os dados geoespaciais à base de dados PostGIS e, devolve o serviço de mapa (WMS) para o DRF, que enfim envia para o Angular.

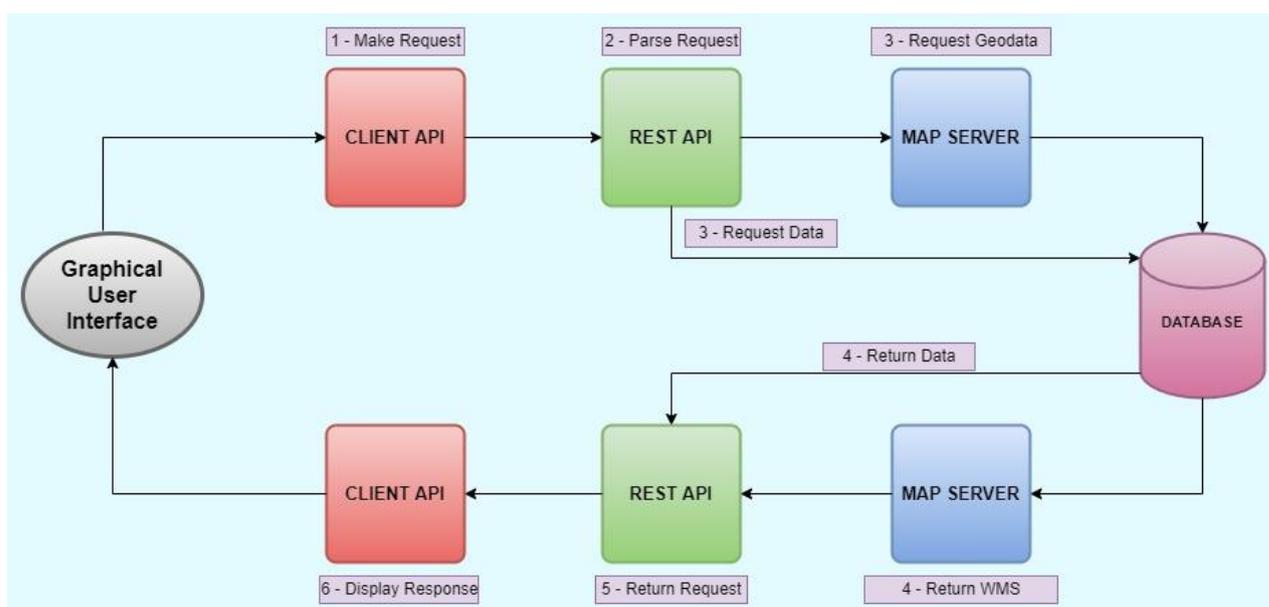


Figura 11: Diagrama sobre como a informação percorrerá dentro do sistema do WebSIG. (Adaptação de CINCOVIĆ *et al.*, 2019).

3.4 Domínio, idioma e *mockup*

Com relação ao domínio e ao nome do projeto na Web, a princípio o WebSIG terá o domínio christianworldmapping.uc.pt, e o seu nome será o Christian World Mapping (CWM), que em poucas palavras descreve os pontos-chaves deste trabalho, o nome da religião e a cartografia, para além de intrinsecamente referenciar a World Christian Database. O idioma de todas as páginas do projeto será o inglês de modo a abranger um maior número de usuários, e que num futuro não muito longe possa existir a opção de seleção de idiomas, a fim de alcançar mais utilizadores.

Pensou-se ainda, no formato do *layout* da página de boas-vindas (Figura 12), contendo uma caixa de texto com informações resumidas sobre o projeto, botões que levarão o utilizador a outras páginas, como:

- ❖ “*About*” – com informações mais detalhadas sobre o projeto;
- ❖ “*Data*” – com informações detalhadas sobre a fonte de dados;
- ❖ “*Chronology*” – com informações sobre os períodos do cristianismo;
- ❖ “*Contact*” – com indicações em como contactar o autor;
- ❖ “*Login*” – área para inserir as credenciais de acesso à página principal.

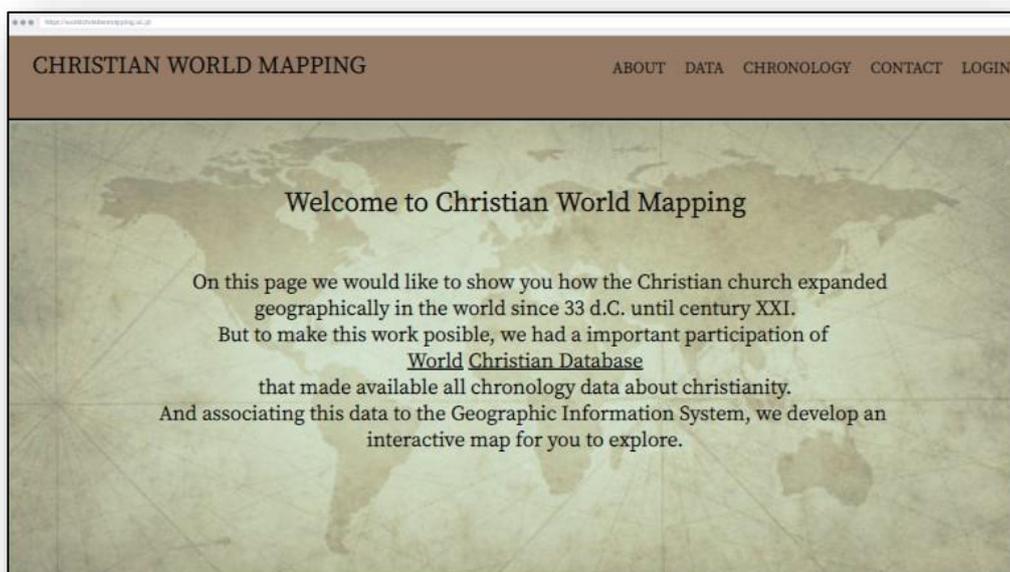


Figura 12: *Mockup* da página de boas-vindas.

As cores escolhidas, num tom mais castanho, remetem a ideia de algo mais histórico, uma forma de envolver o utilizador e, fazê-lo sentir-se num ambiente que contrasta com o tipo de informação que busca.

A página principal foi desenhada (Figura 13), pensando-se em como cada parte do sistema poderia se enquadrar, embora alterações futuras possam surgir. Assim, a página contará com:

- ❖ O cabeçalho, com o nome da página e os botões: “About”, “Data”, “Chronology”, “Contact” e “Logout”;
- ❖ O espaço dedicado para a apresentação do mapa de base, neste mesmo lugar após as seleções das opções disponíveis para o utilizador, aparecerá o mapa para visualização e respectiva legenda;
- ❖ A área separada para inserção dos *dropdown*, permitirá o utilizador escolher as opções vindas da base de dados para cada item e, neste mesmo sítio, abaixo dos *dropdown* haverá o espaço que devolverá o resultado da busca.

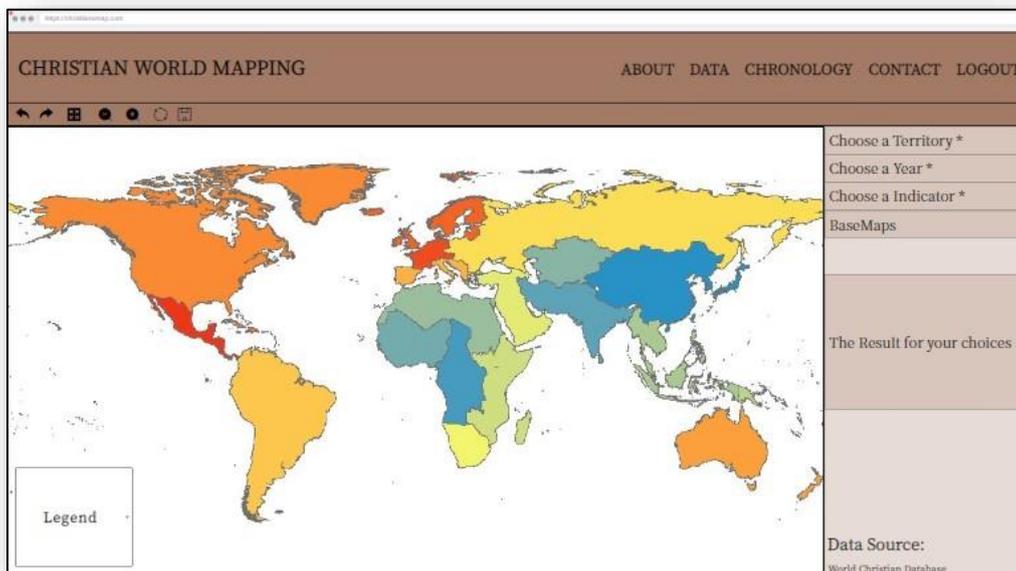


Figura 13: Mockup da página principal do WebSIG.

A título de informação, estes *layouts* foram desenhados através da aplicação Mockplus (<https://www.mockplus.com/>) e os diagramas apresentados anteriormente através da aplicação Diagrams (<https://app.diagrams.net/>).

4. Implementação do sistema

Neste capítulo apresenta-se os processos para o desenvolvimento do CWM, iniciando com a construção do *backend* e da base de dados, a manipulação da informação, a construção do *frontend* e por fim os serviços de mapas. Em cada secção é informado como que as ligações são feitas entre as tecnologias utilizadas.

4.1 Implementação do Django REST framework

Na construção do *backend*, trabalhou-se com os quatro módulos principais do sistema DRF, os “Models”, os “Serializers”, as “Views” e os “URLs”.

A descrição do que foi feito no módulo “Models”, encontra-se na próxima secção, base de dados.

No módulo “Serializers”, utilizando classes em Python para cada tabela da base de dados, designou-se como os dados serão inseridos nela e posteriormente a requisição das informações. O módulo, faz a seriação dos dados, colocando-os num formato mais simples (JSON), para depois serem enviados as “Views”, quando requisitados. Neste ambiente definiu-se a hierarquia para quando o Angular solicitar as informações, seja enviado tudo de uma vez para o *frontend*, isto significa que, o DRF enviará os dados de acordo com a sequência: para cada tipo de território envia-se junto os seus respectivos anos e, dentro dos anos os respectivos indicadores, e por último a respectiva *layer* do tipo de território, do ano e do indicador, possibilitando assim, o filtro da informação. Dessa forma, na prática o que o sistema faz é, quando o utilizador escolhe uma das opções do tipo de território, o próximo *dropdown* apenas mostrará as opções de escolha referente aos anos deste tipo de território, por conseguinte, após a escolha do ano o utilizador no próximo *dropdown* (indicador), apenas visualizará os indicadores daquele ano selecionado, para finalizar quando todas as opções estiverem escolhidas o sistema através do nome da *layer*, devolve o mapa para visualização.

No módulo “Views”, criou-se utilizando as classes em Python e o *package* “viewsets.ModelViewSet” do DRF, as funcionalidades do sistema que, possibilita a listagem dos registos e os métodos CRUD para, os anos, os indicadores, os tipos de territórios, os territórios, as *layers*, e a população (ressaltando que a lista com os tipos de territórios quando requisitada através do método GET, ela leva junto os dados que foram vinculados a eles no “Serializer”). Criou-se ainda, a funcionalidade para buscar o valor único (numérico ou percentual), quando um

território, um ano e um indicador forem escolhidos pelo utilizador. Criou-se mais, as “Views”, utilizando definições em Python, para os serviços de mapa que, fazem os pedidos do WMS e da legenda (uma opção do GeoServer), para cada *layer* indicada. Sem esquecer de mencionar a “View” criada para a autenticação do utilizador, utilizando o *package* Auth do DRF.

Por fim, no módulo “URL’s”, utilizando o *package* “routers” do DRF para os dados alfanuméricos, basicamente o que se fez foi a nomeação da URL e a indicação de sua respectiva “View”. Para os dados geoespaciais, foi preciso criar “URL’s” que identificasse no GeoServer, os locais onde poderia solicitar os serviços de mapas.

Analisando-se o diagrama da Figura 14, que está num maior nível de detalhe para o sistema do DRF, descreveremos sucintamente como a informação circula em cada parte da API (*Application Programming Interface*). Primeiramente, quando o serviço é requisitado pela API do cliente, o módulo “Auth” verificará junto à base de dados se o utilizador tem autorização para aceder as informações do sistema, se tiver autorização segue o pedido para as “Views”, se não tiver, o sistema não devolve nenhum pedido. Quando o pedido for relacionado aos dados alfanuméricos, a “View” do pedido buscará a informação na base de dados, passando pelo “Serializer” e pelo “Models”. Quando o pedido for relacionado com os dados geoespaciais, a “View” correspondente o redirecionará para o GeoServer, que por sua vez, buscará os dados relacionados na base de dados e os devolverá para a View. Posteriormente, as requisições feitas inicialmente são devolvidas para a API do cliente.

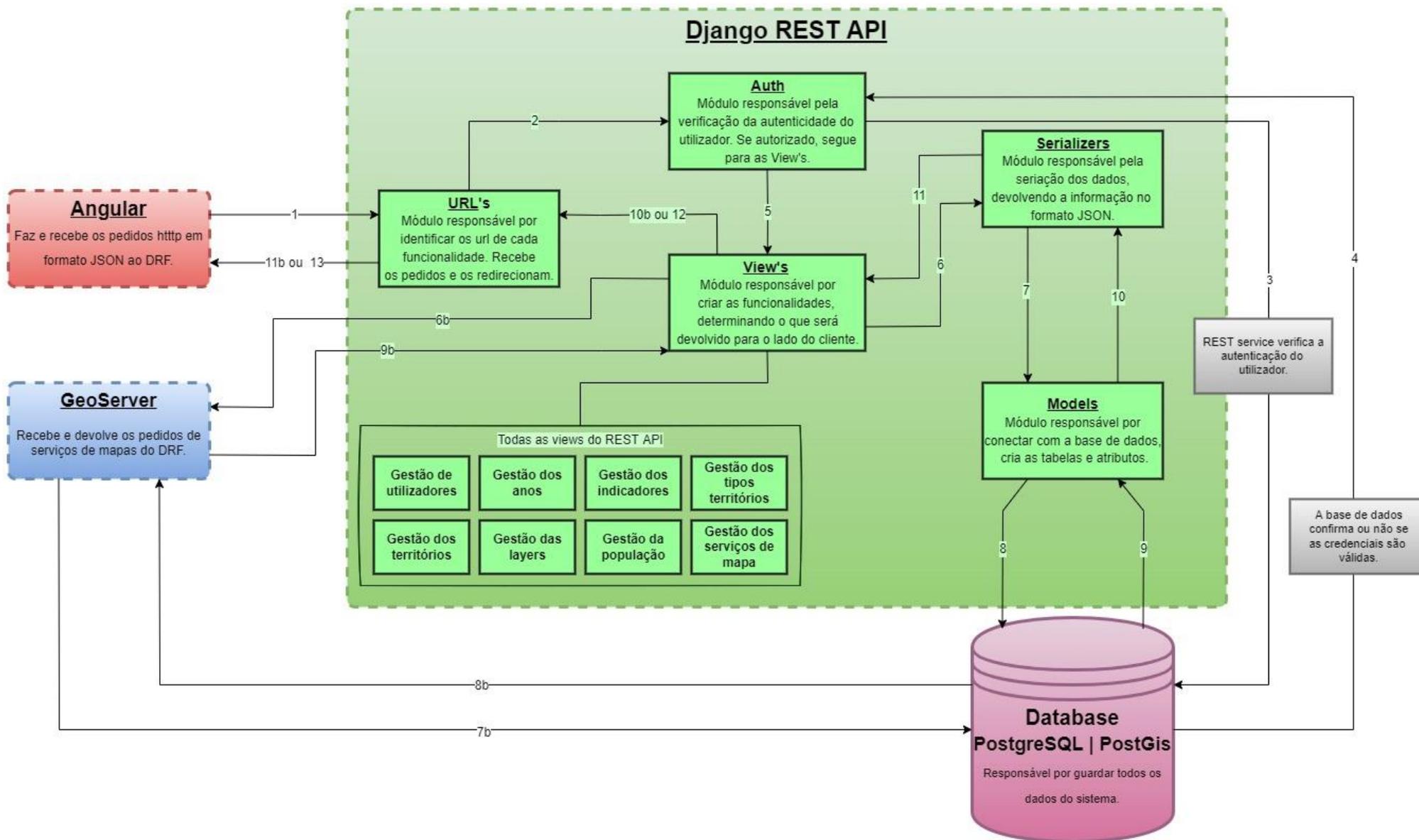
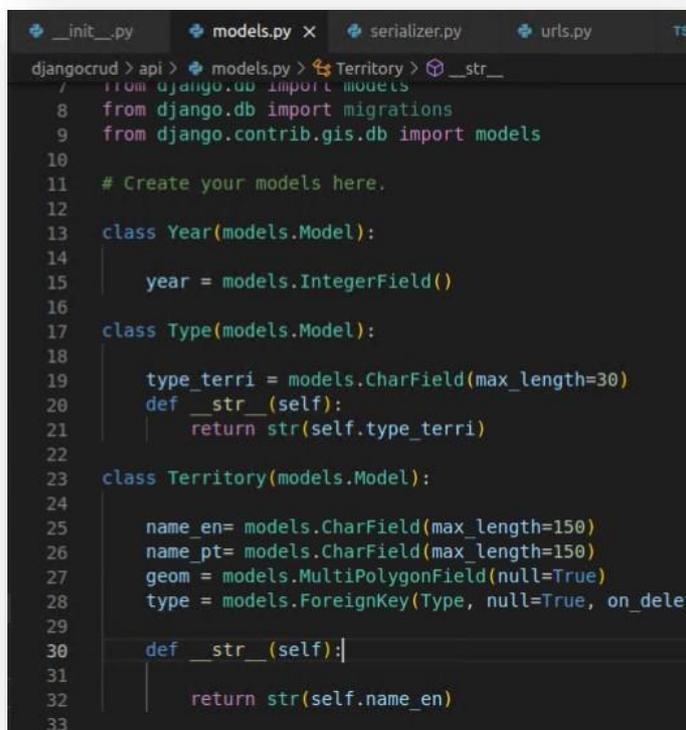


Figura 14: Diagrama detalhado sobre a circulação da informação no DRF.

4.2 Construção da base de dados relacional PostgreSQL | PostGIS

A criação da base de dados fez-se através das classes em Python, por via da componente Models do Django REST framework, onde cada classe criada representa uma tabela e a definição de cada atributo da tabela é estabelecido dentro da classe (Figura 15). Em alguns casos, na criação da tabela precisou-se dizer que determinado atributo era uma chave estrangeira, ao fazer esta afirmação, o servidor REST faz automaticamente a relação entre as tabelas. No caso, da tabela “Type”, existe a ligação de um para muitos, com a tabela “Territory”. Já a tabela “Population” consiste na relação de um para muitos, com as tabelas “Year”, “Indicator”, “Territory” e “Layer”.



```
__init__.py  models.py x  serializer.py  urls.py  TS
djangocrud > api  models.py > Territory > __str__
/
7   from django.db import models
8   from django.db import migrations
9   from django.contrib.gis.db import models
10
11  # Create your models here.
12
13  class Year(models.Model):
14
15      year = models.IntegerField()
16
17  class Type(models.Model):
18
19      type_terri = models.CharField(max_length=30)
20      def __str__(self):
21          return str(self.type_terri)
22
23  class Territory(models.Model):
24
25      name_en= models.CharField(max_length=150)
26      name_pt= models.CharField(max_length=150)
27      geom = models.MultiPolygonField(null=True)
28      type = models.ForeignKey(Type, null=True, on_delete=
29
30      def __str__(self):|
31
32          return str(self.name_en)
33
```

Figura 15: Exemplo de código para criação das tabelas, através das classes em Python.

Utilizando o esquema traçado para construção da base de dados, demonstrado na Figura 5 da secção 3.1.1, veremos a seguir como ficou definido cada tabela e seus campos:

- ❖ Tabela “Year”:
 - Id: identificador da entidade;
 - Year: os anos disponíveis extraídos da WCD.

- ❖ Tabela “Type”:
 - Id: identificador da entidade;
 - Terry_type: os tipos de territórios.

- ❖ Tabela “Territory”:
 - Id: identificador da entidade;
 - Name_en: nome dos territórios em inglês;
 - Name_pt: nome dos territórios em português;
 - Geom: geometria dos polígonos;
 - Type: identificador dos tipos de territórios;

- ❖ Tabela “Indicator”:
 - Id: identificador da entidade;
 - Final_name: nome completo dos indicadores disponíveis;
 - Short_name: nome abreviado dos indicadores disponíveis;
 - Unidade: a unidade de medida (número real ou percentagem).

- ❖ Tabela “Layer”:
 - Id: identificador da entidade;
 - Workspace: nome da área de trabalho que estará a *layer*;
 - Store: nome do local onde estará a *layer*;
 - Layer: nome da *layer*.

- ❖ Tabela “Population”:
 - Id: identificador da entidade;
 - Year: identificador da tabela “Year”;
 - Territory: identificador da tabela “Territory”;
 - Indicator: identificador da tabela “Indicator”;
 - Layer: identificador da tabela “Layer”;
 - Values: valores de cada registo para cada combinação (um ano, um identificador e um território);

Deste modo, esta definição da estrutura da base de dados relacional (Figura 16), possibilitará no futuro uma maior flexibilidade quanto a inserção de novos dados.

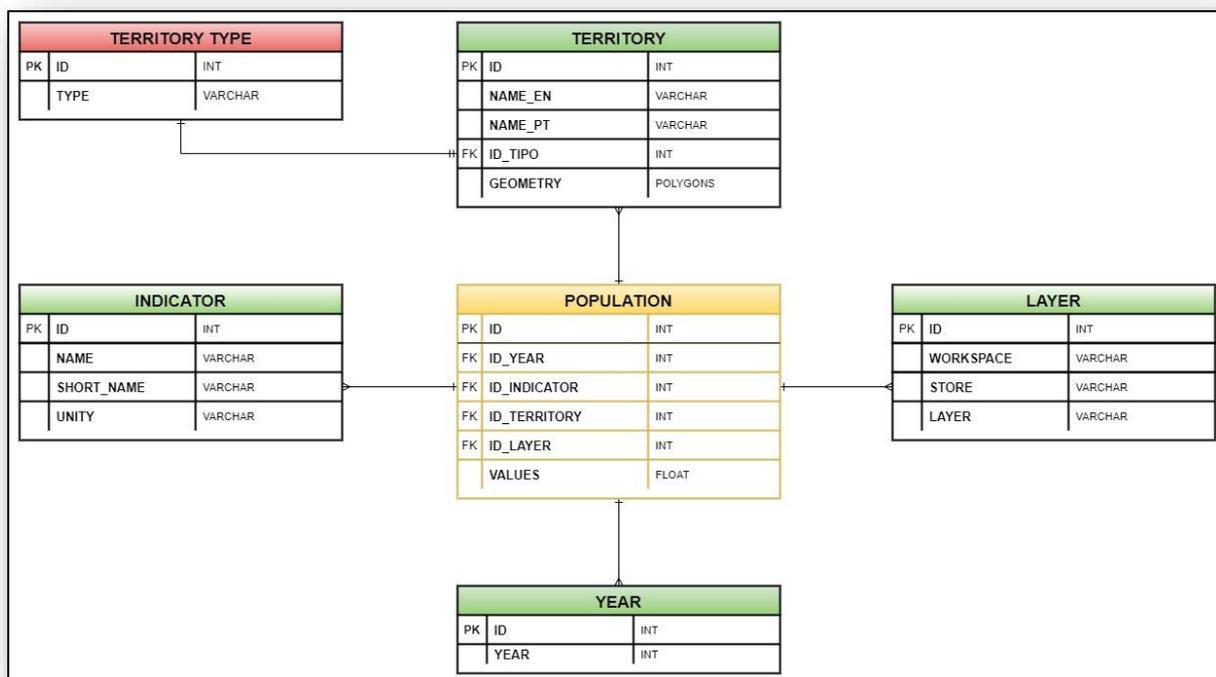


Figura 16: Estrutura da base de dados relacional.

4.3 Manipulação dos dados e criação dos WMS

Por meio de *scripts* Python, utilizando o Jupyter Notebook, foi possível conectar-se com a base de dados utilizando o identificador “URL” de cada “View” e, importando os módulos “requests” (para aplicar os métodos CRUD), “json” (definição do formato dos dados) e “IPython.display” (para mostrar o que foi executado). Por conseguinte, criou-se ficheiros no Jupyter Notebook para a gestão dos dados de cada tabela. A título de exemplo, no Anexo D encontra-se as linhas de código utilizadas para inserir e manipular os dados geoespaciais relacionados aos territórios, para as demais gestões dos dados os códigos estarão disponíveis na página do GitHub do autor².

Ressalta-se que para a importação de dados provenientes de ficheiros Excel (xls) ou ficheiros SHP, os campos destas tabelas foram os mesmos campos das tabelas da base de dados, para que a informação fosse inserida corretamente.

² <https://github.com/yurimtig>

Para a tabela “Type”, cada registo foi criado manualmente através da página da “Administration” do próprio DRF, por se tratar de apenas quatro registos.

As tabelas “Year” e “Indicator”, os dados foram inseridos através da importação de ficheiros Excel (xls) com os respectivos valores para cada tabela.

A tabela “Territory”, por sua vez, utilizou-se dois caminhos para inserção dos dados, o primeiro, teve de se criar um script no próprio DRF, para importar do diretório do Linux os dados geoespaciais em formato SHP para os “Países”, usando-se o *package* LayerMapping, proveniente do próprio Django REST. O segundo caminho tomado destinou-se a manipulação dos dados geoespaciais relacionados aos “Continentes” e às “Regiões do mundo”, utilizou-se para o efeito a gestão de territórios, ficheiro criado no Jupyter Notebook, que possibilitou a aplicação dos métodos CRUD, os dados para estes três tipos de territórios também estavam em formato SHP. A diferença entre os caminhos seguidos se deu pelo facto de que os dados com as informações sobre os países, não estavam sendo exatamente inseridos na base de dados, faltava-se alguns países, por isso encontrou-se a alternativa do módulo LayerMapping.

Para a tabela “Layer”, criou-se as listas com os anos e indicadores, para em seguida, criar-se os ciclos (*loops*), estes por sua vez, inserem as informações na base de dados relativas ao nome do “Workspace” e “Store” que sempre serão os mesmos e, o nome da *layer*, definido por indicador, ano e tipo de território na base de dados. Destaca-se que o nome destes registos será idêntico aos criados no GeoServer.

Na manipulação dos dados para a tabela de relacionamento “Population”, importou-se do diretório o ficheiro Excel (xls) com todos os dados extraídos da WCD com os valores (numéricos ou percentuais) para cada indicador, de acordo com o ano e território. Criou-se a lista com os nomes das colunas do ficheiro relativos aos indicadores para que quando o código chamar os valores, busque somente os que estão na coluna assinalada. Criou-se a lista com o número do identificador de cada registo do ficheiro, para posteriormente ser identificado e associado ao identificador da tabela “Territory” indicando que o valor do registo será relacionado com o território e, criou-se ainda, a lista com os identificadores da tabela “Layer”. Utilizando três ciclos, foi possível seleccionar os valores certos segundo o ano, o território e o indicador, que basicamente o que se fez foi, para cada ano tem um indicador, e para cada indicador tem o território e, para cada território tem um valor correspondente. Após inúmeros testes para verificar a correta execução do código, se os valores estavam realmente associados a cada identificador correspondente, o procedimento foi executado no final para cada um dos tipos de território, gerando um total de 46524 registos na tabela “Population” da base de dados.

A seguir o preenchimento de cada uma das tabelas na base de dados com as informações sobre o cristianismo e com os dados geoespaciais, criou-se as *layers* no GeoServer para posteriormente serem apresentadas no Angular, quando forem requisitadas. Como mencionado outrora o *workspace*, *store* e a *layer* no GeoServer receberam os mesmos nomes que estão nos registos da tabela “Layer”, isto é para quê, quando houver um pedido do utilizador, estes nomes sejam facilmente identificados pelo sistema na busca pelo mapa relacionado.

Prosseguindo, utilizou-se a biblioteca “geo.Geoserver”, para automatizar o processo de criação do *Workspace*, *Store* e de cada *Layer* e o carregamento do *Style* no GeoServer, sendo assim, para:

- ❖ O *Workspace* foi preciso invocar a função “geo.create_workspace” e preencher o parametro de entrada: nome do *workspace*;

- ❖ O *Store* foi preciso invocar a função “geo.create_featurestore” e preencher os parâmetros de entrada:
 - Nome do *store*;
 - Nome do *workspace*;
 - Nome da base de dados;
 - Nome do host;
 - Nome do utilizador da base de dados;
 - Senha do utilizador da base de dados.

- ❖ As *Layers*, através de ciclos (Figura 17), invocando a função “geo.publish_featurestore_sqlview” e preenchendo os parâmetros de entrada:
 - Nome do *store*;
 - Nome da *layer*;
 - Query (em SQL) para buscar na base de dados as informações (alfanumérica e geoespaciais) e determinar a criação de cada *layer*;
 - Nome da coluna principal;
 - Nome do *workspace*.

- ❖ O carregamento dos *Styles* através de ciclos, invocando a função “geo.upload_style” (para importar os ficheiros .sld) e preenchendo os parâmetros:
 - Com a indicação do caminho do diretório onde encontra-se os ficheiros .sld;
 - Nome do *workspace*.

- ❖ A publicação dos *Styles*, através de ciclos, invocando a função “geo.publish_style” e preenchendo os parâmetros:
 - Nome da *layer* que será associado o *style*;
 - Nome do *style* anteriormente importado para o sistema do GeoServer;
 - Nome do *workspace*.

```

for x in year:
    for y,z in zip(indicadores, indiname):
        sql = "SELECT year, final_name, name_en, name_pt, values, geom "\
              "FROM api_year "\
              "INNER JOIN api_populations "\
              "ON api_year.id = api_populations.year_id "\
              "INNER JOIN api_indicator "\
              "ON api_indicator.id = api_populations.indicator_id "\
              "INNER JOIN api_territory "\
              "on api_territory.id = api_populations.territory_id "\
              "WHERE "\
              f"year = '{x}' AND "\
              f"final_name = '{y}' AND "\
              "type_id = '4'"
        geo.publish_featurestore_sqlview(store_name = st, name = (z + '-' + x + '-country'),
                                         sql = sql, key_column = 'year', workspace = ws)

```

Figura 17: Exemplo de código, utilizando as linguagens de programação Python e SQL.

Os ficheiros sld foram criados para os tipos de territórios: “Continentes”, “Regiões do mundo” e “Países”, de forma automática através de códigos em Python e armazenados na máquina virtual. Utilizou-se o critério quantitativo para distinguir as cores dos polígonos de cada *layer*, criação das classes, recorrendo ao método Jenks Natural Breaks (por meio da biblioteca Python “jenkspy.jenks_break”), que cria os grupos de acordo com a proximidade dos valores, permitindo uma melhor distribuição destes valores de acordo com o número de classes. Elas por sua vez, foram determinadas de acordo com o tipo de território, por exemplo, para os “Continentes” definiu-se 4 classes, para as “Regiões do mundo” do ano 33 ao ano de 1850 foram 5 classes e do ano de 1900 até o ano de 2050 foram 10 classes e, para os “Países” foram definidas 10 classes.

Para os continentes poderia empregar o critério de classificação categorizado, que permitiria que eles fossem distinguidos pelo nome do continente e não pelos valores de cada um,

se assim fosse, não seria possível comparar visualmente por exemplo, qual o continente com o maior ou menor valor, ou os com valores aproximados. Acrescenta que as cores sempre seriam as mesmas para cada continente, independente da escala temporal ou dos indicadores.

Para finalizar, com relação as cores dos mapas, após inúmeros testes para identificar qual a rampa de cores teria a melhor representação, optou-se pela rampa de cores no tom castanho, que por sua vez, melhor se associa ao fator população.

4.4 Implementação do Angular

Na API do Angular, após as configurações necessárias do sistema, criou-se as funções no módulo “Service” da aplicação, indicando as “URL’s” de cada “View” do Django REST framework, estabelecendo a comunicação entre eles.

Para cada página do CWM, criou-se uma “Component”, composta por três formatos de arquivos, o HTML, o CSS e o TypeScript, sendo estas “Component’s” ligadas a “Component” principal da API. Assim, as páginas criadas foram:

- ❖ “Home page” – página inicial da aplicação;
- ❖ “About” – página que descreve sucintamente todo este projeto;
- ❖ “Data Source” – página que descreve a fonte de dados e editores;
- ❖ “History” – página que descreverá a história do cristianismo (em construção);
- ❖ “Contact” – página de envio de mensagens ao autor;
- ❖ “Login” – página para inserir as credenciais de acesso;
- ❖ “Mapping” – página principal do sistema.

Estas páginas estão todas interligadas, e a navegação por entre elas é possível e aberta ao utilizador, exceto, a página principal que requer autorização.

Considerando a página com a maior iteração de informações, a página principal, a seguir veremos as principais funções implementadas em TypeScript que possibilitaram alcançar os objetivos iniciais previstos. O código destas funções não será detalhado aqui, mas estará disponível para o leitor na página do GitHub do autor³. Portanto, as quatro principais funções utilizando os serviços Web, são:

³ <https://github.com/yurimtig>

- ❖ GetAll – faz a busca pela “View” dos tipos de territórios, que quando requisitada envia toda a informação da base de dados (tipos de territórios, anos, indicadores e *layers*) para o *frontend*, para posteriormente serem separadas.
- ❖ AddLayer – Em efeito cascata, após a seleção do tipo de território, do ano e do indicador pelo utilizador na página principal, é fornecido o registo da “Layer”. Com esta informação a função envia o pedido de mapa ao DRF que, por sua vez, solicita ao GeoService o serviço de mapa para estes dados. Após recebe o WMS, a função o adiciona ao *basemap*. Quando um novo serviço é requisitado, a função apaga este WMS e insere um novo, evitando as sobreposições de mapas.
- ❖ Selected – recolhe quando seleccionado pelo utilizador, o número do identificador do ano, do indicador e do território para posteriormente requisitar à base de dados o valor (numérico ou percentual) corresponde a estes fatores.
- ❖ Basemaps – utilizando a biblioteca do Leaflet, e os basemaps da ESRI, esta função permite visualizar na página o mapa do mundo, de acordo com a escala pré-definida. Conforme o utilizador aumenta ou diminui a escala (zoom in e zoom out) do mapa, ele automaticamente ajusta a imagem através do “Tile Layer” da Open Street Map, invocado pelo Leaflet.

Para concluir o capítulo, o diagrama da Figura 18 apresenta a estrutura do sistema indicando com detalhe como a informação circula dentro de todo o sistema da CWM. Recordando que durante cada fase de implementação do sistema foi falado em como o *backend*, o *frontend*, o GeoServer e a base de dados são conectados, contudo, é importante frisar estas ligações. Desse modo, recorrendo a Tabela 10, utilizou-se a identificação da porta de cada tecnologia, quando ela fosse requisitada. O Django REST framework, faz a gestão destas portas, indicando onde guardar os dados, onde requisitar os serviços de mapas e para onde enviá-los quando fossem solicitados.

5. Resultados

Este capítulo destina-se à apresentação dos resultados da implementação do projeto CWM, que foi desenvolvido ao longo de alguns meses. Por se tratar de algo visual, apresenta-se dessa forma os resultados através de figuras, destacando as três principais páginas: a de boas-vindas, a de login de acesso e a principal com o mapa e as opções de escolha.

Na página inicial do projeto, a de boas-vindas ao utilizador (Figura 19), é exibido uma mensagem de texto, dando as boas-vindas e um pequeno resumo do projeto CWM referenciando a fonte dos dados. Nesta página foi inserido um mapa do mundo de fundo, para ambientar o utilizador com o tema em questão. No mesmo propósito as cores e o formato da fonte das letras, remetendo para algo histórico, seguindo o que se havia planejado anteriormente.

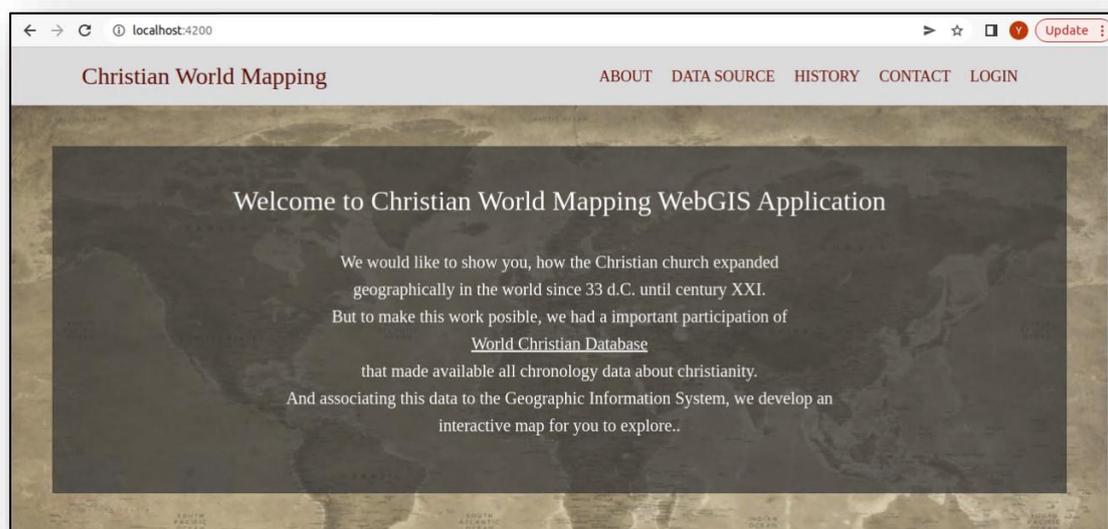


Figura 19: Página de boas-vindas do Christian World Mapping.

Nesta página foram criados os botões: “About”, “Data Source”, “History”, “Contact” e “Login”, quando algum deles é clicado, o utilizador é redirecionado para a respectiva página. Esta estrutura da página inicial se mantém para as páginas “About”, “Data Source” e “Contact”. Já nas páginas “History”, “Login” e principal mantém-se apenas o cabeçalho e os botões.

A página de Login (Figura 20), é destinada para a inserção das credenciais do utilizador (*username* e *password*), após o sistema confirmar que o utilizador existe e que a *password* está correta o sistema redireciona a aplicação de cliente para a página principal.

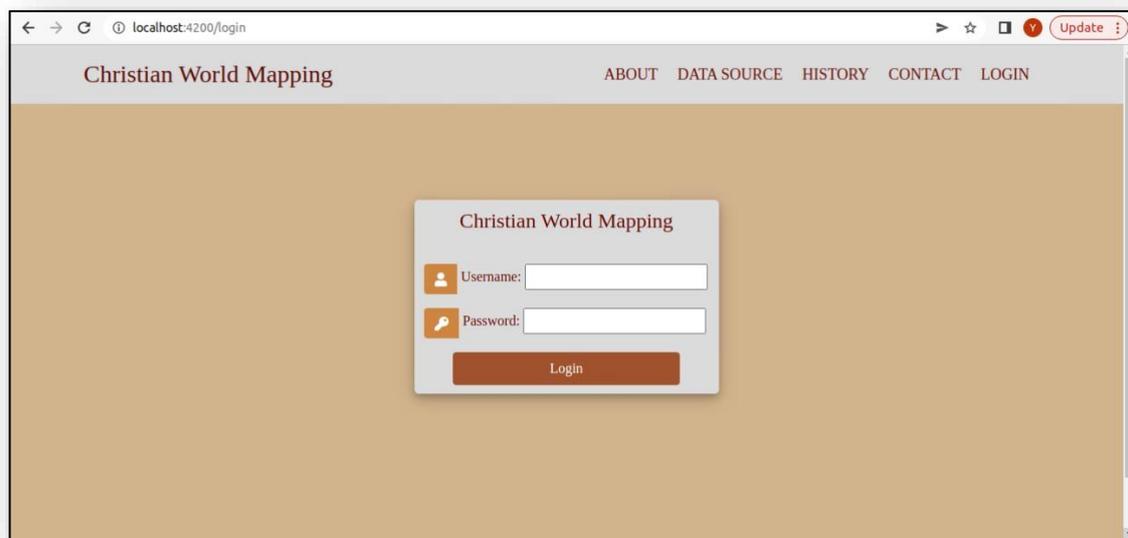


Figura 20: Página de login da Christian World Mapping.

Por último, a página principal: apresentando o cabeçalho com as opções para dirigir-se a alguma outra página do CWM; abaixo do cabeçalho o mapa de base *National Geographic* da ESRI, que sempre que a página for atualizada o mapa estará nesta escala. Podendo o utilizador fazer as alterações de escala, quando fizer zoom in ou zoom out e, arrastar o mapa de um lado para o outro quando permanecer com o click do rato pressionado. Ao lado direito encontram-se três *dropdowns* de seleção obrigatória para o surgimento do mapa (tipos de territórios, os anos e os indicadores). Para além destes há um quarto *dropdown* opcional que permiti seleccionar o território e logo visualizará o valor que este apresenta para um determinado indicador e ano. Assim, como mostra a Figura 21, a página principal ficou com as componentes distribuídas, de modo a proporcionar maior interação com o utilizador.

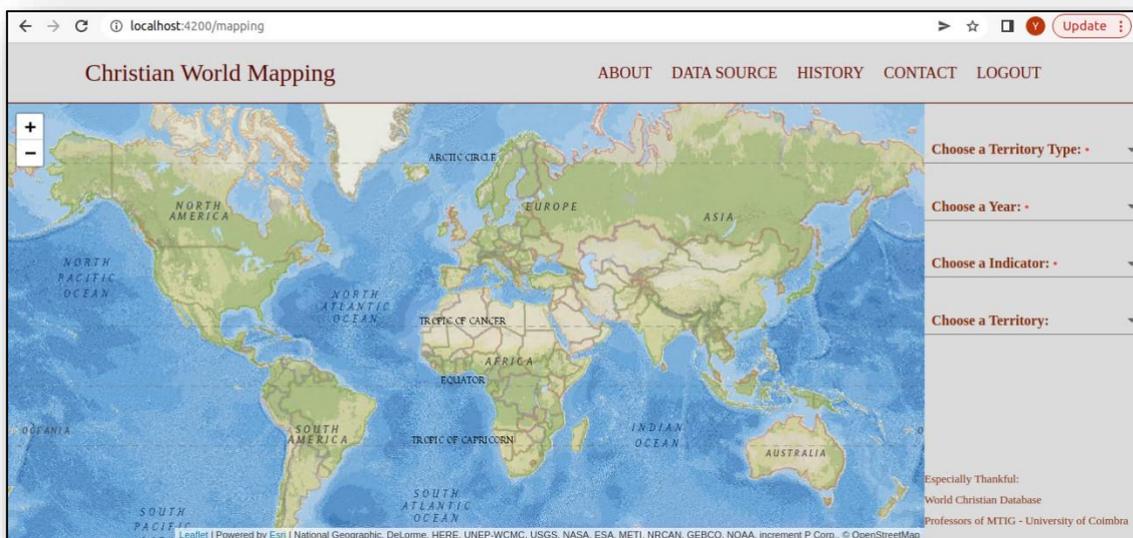


Figura 21: Página principal do WebSIG Christian World Mapping.

A título de exemplo de como ficaram os *dropdowns*, a Figura 22 demonstra as opções para o tipo de território, para os anos deste tipo de território escolhido e os indicadores de acordo com as seleções anteriores.

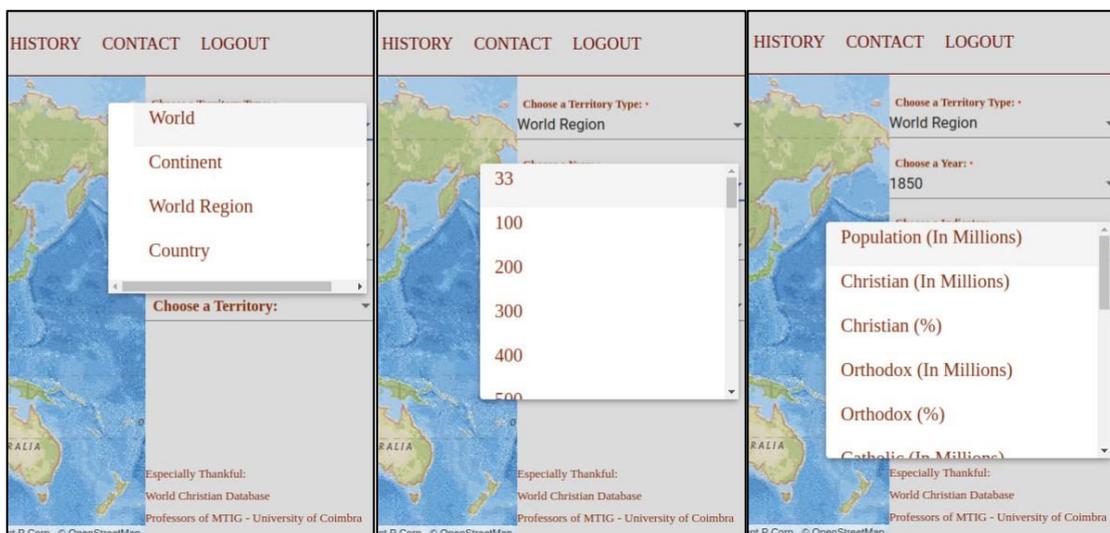


Figura 22: Demonstração das informações contidas nos três *dropdowns* de escolhas obrigatórias.

Caso o utilizador seleccione o tipo de território mundo, o que o sistema lhe apresentará serão as informações numéricas referente ao ano e indicador escolhidos.

Se a escolha do utilizador for o tipo de território “continente”, o sistema lhe apresentará o mapa com os continentes, com a rampa de cores de acordo com os valores para aquele ano e para aquele indicador (Figura 23).

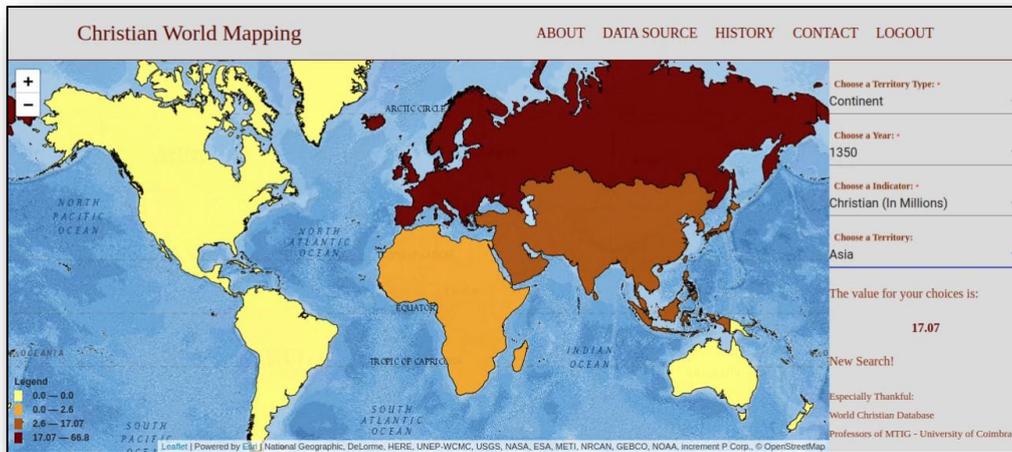


Figura 23: Resultado para os “Continentes”, no ano de 1350 e o indicador o nº de cristão.

Se a escolha do utilizador for o tipo de território “Regiões do mundo”, o sistema lhe apresentará o mapa com as “Regiões do mundo”, com a rampa de cores de acordo com os valores para aquele ano e para aquele indicador. Na Figura 24, observa-se por exemplo, a forte influência no período do Império Romano, a partir do imperador Constantino que oficializou o cristianismo como a religião do império (Hurlbut, 1967), expandindo a religião para o Sul da Europa, Norte da África e Médio Oriente.

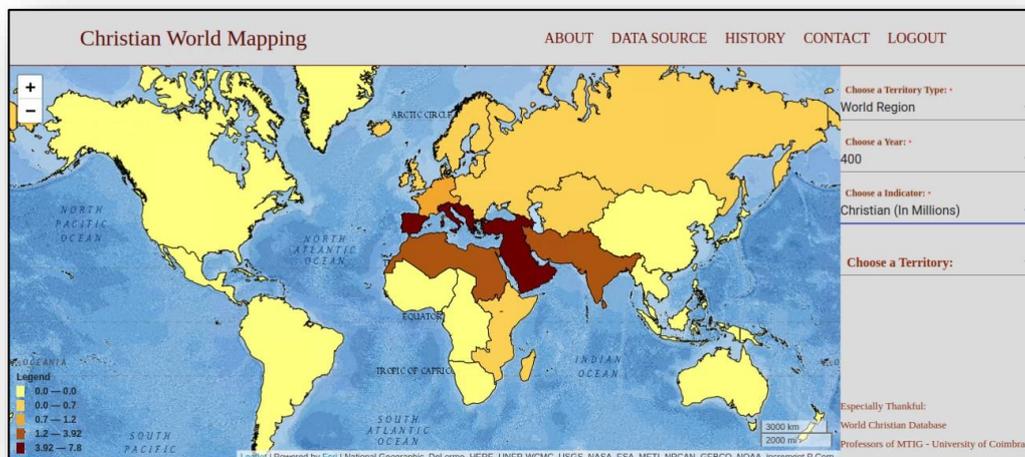


Figura 24: Resultado para as “Regiões do mundo”, o ano de 400 e indicador o nº de cristão.

Outro período que se pode destacar é o das Grandes Navegações, a partir do século XV, expandindo a religião para outros continentes, uma busca no CWM para o ano de 1700 por exemplo (Figura 25), nota-se o crescimento do cristianismo na América do Sul, na África Oriental, parte do Sudeste Asiático e Ásia Ocidental.



Figura 25: Resultado para as “Regiões do mundo”, o ano de 1700 e indicador o nº de cristão.

Se a escolha do utilizador for o tipo de território “Países”, o sistema lhe apresentará o mapa com os países da atualidade, com a rampa de cores de acordo com os valores para aquele ano e para aquele indicador. A título de exemplo, e em termos comparativos, a Figura 26 mostra os países com o número de cristãos que seguem o catolicismo; já a Figura 27 apresenta o número de cristãos que seguem o protestantismo e por último a Figura 28 mostra o número de cristãos que seguem o ortodoxíssimo, todos para o ano de 2020.

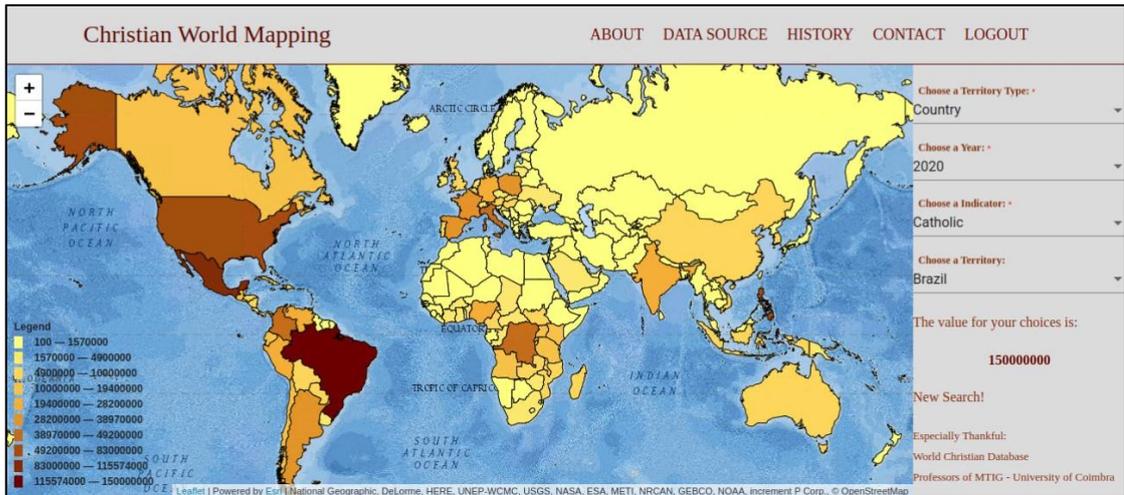


Figura 26: Resultado para “Países”, no ano 2020 e o indicador o número de católicos.

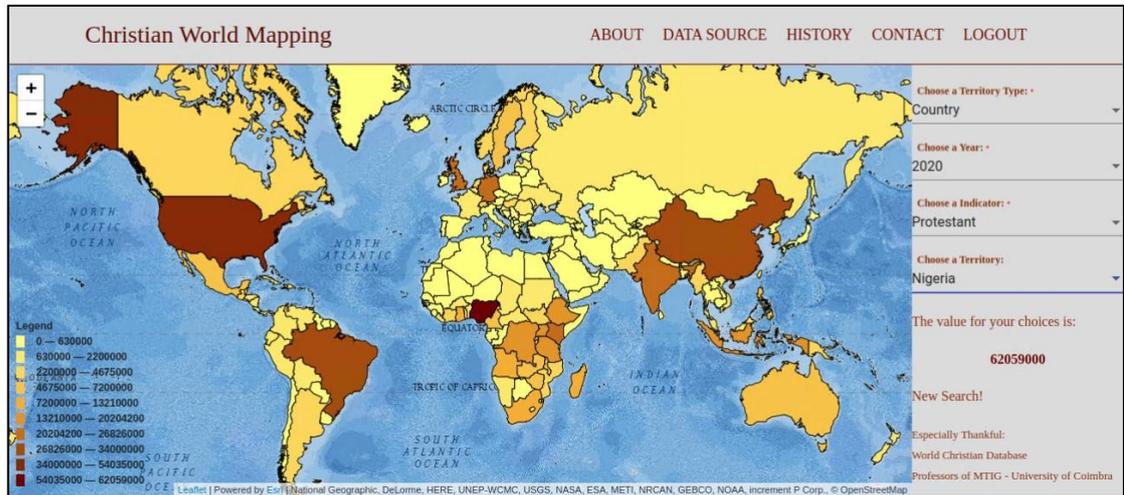


Figura 27: Resultado para “Países”, no ano 2020 e o indicador o número de protestantes.

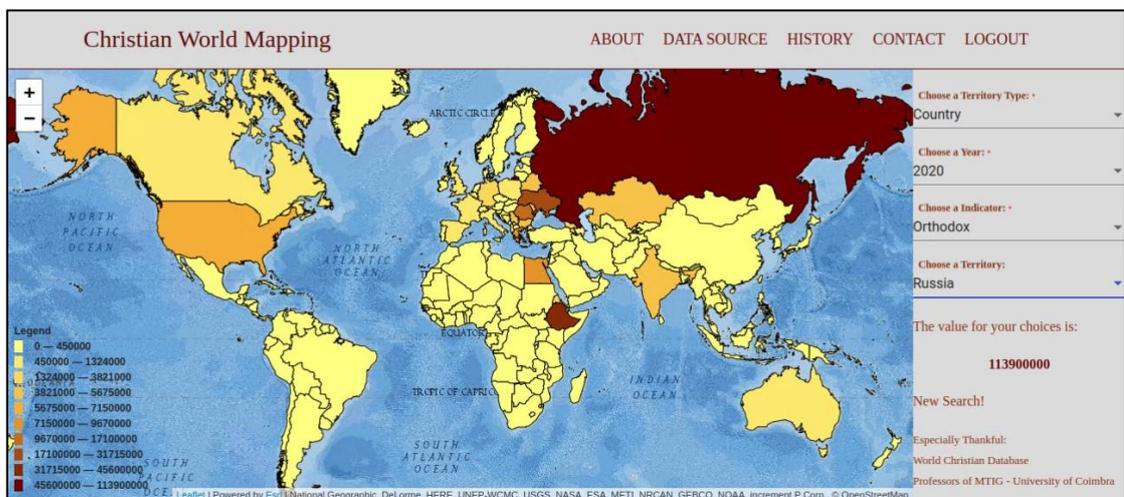


Figura 28: Resultado para “Países”, no ano 2020 e o indicador o número de ortodoxos.

6. Conclusão

Com o intuito de estabelecer uma ligação entre o cristianismo e os Sistemas de Informação Geográfica, surgiu a ideia da criação de um WebSIG que representasse a expansão geoespacial da religião ao longo dos tempos. Em virtude disto, propôs-se a construção de um sistema WebSIG, que viesse apresentar informações alfanuméricas e geoespaciais ao utilizador. Observou-se que não havendo em nenhum outro lugar algo semelhante e, sendo estas informações relevantes a ponto de vista académico, religioso, histórico e tecnológico, procurou-se desta forma o desenvolvimento do WebSIG.

Entretanto, no início havia dois desafios. O primeiro, falar sobre fatos que ocorreram durante a história num contexto geográfico, sabendo que mudanças nos limites territoriais foram se alterando conforme o avanço temporal, isso seria um trabalho árduo e que levaria tempo para cartografar todas estas alterações geoespaciais. O outro desafio que poderia implicar anos de estudos seria, quantificar a população mundial e a população pelas “Regiões do mundo” em distintos períodos da história e para além, a quantificação de uma parcela desta população, neste caso os grupos ligados ao cristianismo. Após buscas, foram encontradas as respostas para estes desafios através da instituição WCD, que trabalha exclusivamente com pesquisas relacionadas à demografia da religião. A WCD, forneceu os dados com a cronologia do número de cristãos desde o seu surgimento até os dias atuais e uma projeção para o futuro, separados numa escala temporal variada e, por tipos de regiões (“Mundo”, “Continentes”, “Regiões do mundo” e “Países”). Com esta solução, este projeto ganhou “alma” e pôde-se então avançar para a etapa seguinte, as tecnologias.

Assim, após a aquisição dos dados e no decorrer do projeto pensou-se em como poderia ser construída a base de dados relacional, e como seria as relações entre as tabelas dos dados alfanuméricas e dos dados geográficos. Pensou-se ainda em, como construir um *backend* utilizando o Django REST framework (DRF), que usa a linguagem de programação Python. E como seria a construção do *frontend* através do Angular, que utiliza as linguagens de programação HTML, CSS e TypeScript. E por fim, como seria a manipulação dos dados utilizando *scripts* Python. Este processo de arquitetar o sistema permitiu uma maior confiança quando se passou para a fase de implementação, constatando a importância do planejamento na construção de um WebSIG.

Deste modo, com a arquitetura do sistema desenvolvida seguiu-se para a construção da base de dados relacional através do próprio DRF, utilizando a base de dados PostGIS, extensão

da PostgreSQL, criando as relações entre os dados alfanuméricos referentes ao cristianismo e os dados geográficos, com a informação em formato vetorial dos quatro tipos de territórios. As funcionalidades do sistema, foram também criadas utilizando o DRF, possibilitando a aplicação do método CRUD (*Create, Read, Update e Delete*) para a inserção e manipulação dos dados à base de dados. Portanto, utilizando as tecnologias citadas, foi possível construir todo o sistema do WebSIG.

No começo, a ideia consistia em ter uma simples página Web que representasse a expansão do cristianismo ao longo dos tempos, onde o utilizador pudesse escolher um ano numa *checkbox* e o sistema devolveria um mapa com a representação. Contudo, com o passar dos meses, essa ideia foi aprimorada chegando à conclusão de que as informações poderiam ser mais bem distribuídas, separando-as por grupos, a fim de possibilitar uma maior flexibilidade quando houvesse uma necessidade de alteração na base de dados e não só, permitir uma variedade de opções a serem escolhidas pelo utilizador, inserindo então os *dropdowns*.

Com base no que foi apresentado, criou-se a interface gráfica do utilizador com a página de boas-vindas ao utilizador; a página de login para ele inserir as credenciais de acesso às informações; a página principal com o mapa de base e os *dropdowns* para escolha das opções apresentadas; a página sobre este projeto; a página de contactos; a página com informações sobre a fonte dos dados; e a página com o espaço dedicado a falar sobre a história do cristianismo, que devido ao curto espaço de tempo não foi concluída mas, estará em construção.

Por conseguinte, aquilo que foi imaginado inicialmente, criar um WebSIG representando a expansão do cristianismo, foi ganhando forma semana após semana. Passando pela fase conceptual, pensando em cada parte do processo e, posteriormente a etapa da implementação, o desenvolvimento dos códigos e suas conexões. Contudo, não foram só acertos durante a criação do sistema, inúmeros erros apareceram e tiveram de ser corrigidos, para que por fim, os objetivos iniciais previstos pudessem ser aprimorados e alcançados.

Portanto, o desenvolvimento deste projeto possibilitou uma elevada compreensão teórica e prática, no que concerne, as tecnologias *open source* utilizadas e, à relação entre estas tecnologias que possibilitaram a criação do WebSIG, Christian World Mapping.

6.1 Trabalho futuro

Devido ao tempo limitado, não foi possível avançar em algumas partes deste projeto, assim, pretende-se para os próximos meses, a finalização da página “History” do CWM, inserindo os principais fatos ocorridos na história do cristianismo.

Tenciona-se a criação do gráfico de barras na página principal, com as informações referentes aos valores de todos os indicadores para o tipo de território e o ano escolhido, possibilitando uma comparação dos dados pelo utilizador.

Um outro passo extremamente importante a ser dado, será o pedido oficial à World Christian Database, para a abertura dos dados para todos os utilizadores, ou seja, colocar o CWM num servidor e não impor restrições de acesso, isto porque, atualmente na página Web da instituição só acedem aos dados utilizadores que tenham autorizações, dessa forma, para não sofrer nenhum embargo ou problemas futuros será feito este pedido. Nas duas possibilidades de resposta por parte deles, a primeira sendo positiva, o projeto avança para a fase de implementação do sistema no servidor da Universidade de Coimbra. Caso a resposta seja negativa, o projeto a princípio ficará como trabalho académico para conclusão do curso do MTIG, não descartando a possibilidade de se encontrar dados abertos para o tema futuramente, ainda que sejam apenas para os séculos XX e XXI, ou seja, esta possibilidade permitiria inserir apenas os dados mais recentes.

Objetiva-se mais, a criação da página de registo do utilizador, com o preenchimento dos campos, *username*, e-mail, *password* e confirmação de *password*, para posteriormente poder aceder as informações da página principal.

Por conseguinte, serão os próximos meses destinados a implementação do CWM no servidor da UC, possibilitando a ativação dele na Web. Para isso, mais conhecimentos técnicos deverão ser adquiridos.

Após a ativação do CWM, buscará por apoiadores que contribuirão com a divulgação do WebSIG. Como a Sociedade Bíblica de Portugal, Sociedade Bíblica Brasileira, a própria WCD, a Aliança Evangélica Portuguesa e representantes da igreja Católica em Portugal.

Pretende-se mais para o futuro, a implementação da *layer* com as informações geográficas e alfanuméricas, referente à perseguição que a igreja cristã vem sofrendo no mundo, indicando os países onde ocorrem este fenómeno. A título informativo, no ano de 2020 foram cerca de 340 milhões de cristãos perseguidos (Guerra, 2021), grande parte destes casos ocorreram em países comunistas, países onde o islamismo e o hinduísmo têm maior domínio ou em países que sofrem com a guerra civil, de acordo com a organização A Voz dos Mártires, 2021.

Objetiva-se, investigar quais os procedimentos jurídicos necessários para registo do WebSIG; identificar as leis portuguesas e europeias que regulamentam esta atividade; e os custos que serão inerentes para que este WebSIG possa permanecer online, após o desvinculo com servidor da UC.

Assim, após este tema inusitado implementado num WebSIG, no futuro, quais outros temas poderíamos ver representados num WebSIG?

6.2 Consideração final

Ao longo dos últimos 9 meses, houve dedicação para aprender sobre as tecnologias que permitiram construir o WebSIG, buscando conhecimentos extracurriculares que possibilitassem o desenvolvimento de cada etapa do processo.

Apesar de todas as adversidades encontradas pelo caminho, as barreiras de conhecimentos técnico relacionadas às tecnologias, o curto espaço de tempo para encontrar respostas para cada problema surgido e, conciliar diversas atividades semanalmente. Foi possível aprender e observar o grande progresso no desenvolvimento pessoal, a nível de conhecimentos tecnológicos através do resultado obtido. Estes conhecimentos retidos, serão o impulso necessário para a continuação da aprendizagem no que diz respeito a este “mundo” das tecnologias.

Portanto, após toda dedicação o grande desejo é que este projeto permaneça na Web por tempo indeterminado, sendo gerido e atualizado quando fosse preciso. Contribuindo ativamente com todos aqueles que buscam por informações referentes ao cristianismo, num contexto geoespacial.

Bibliografia

- A VOZ DOS MÁRTIRES. (2021). <https://www.vozdosmartires.com/a-perseguiacao/>. Acedido a 14 de julho de 2021.
- AZURE. (s.d.). *O que é uma base de dados relacional*. <https://azure.microsoft.com/pt-pt/overview/what-is-a-relational-database/#whatis>
- BÍBLIA. (2009). João Ferreira de Almeida. 4ª edição. Sociedade Bíblica do Brasil.
- BRILL (2021). Página da editora Brill. <https://brill.com/page/AboutMain/about> acedida em 03 de julho de 2021.
- BONFIOLI, Guilherme Ferreira. (2006). *Banco de Dados Relacional e Bancos de Dados Objeto-relacional: Uma comparação usando PostgreSQL*. Monografia de graduação – Universidade Federal de Lavras, _____ pág. _____ 6.
http://repositorio.ufla.br/bitstream/1/8354/1/MONOGRAFIA_Banco_de_dados_relacional_e_objeto_relacional_uma_compara%C3%A7%C3%A3o_usando_postgresql.pdf
- BOHRER, Márcio (2014). *Quais são os softwares livres mais utilizados?*
<https://www.oficinadanet.com.br/post/11012-quais-sao-os-sofware-livres-mais-utilizados>
- CAMPOS, Augusto. (2006). *O que é software livre*. BR-Linux. Florianópolis, março de 2006.
<http://br-linux.org/linux/faq-softwarelivre>. Acedido a 10 de julho de 2021.
- CINCOVIĆ, Jelica. DELČEV, Sanja. DRASKOVIC, D. (2019). *Architecture of web applications based on Angular Framework: A Case Study*. <https://www.semanticscholar.org/paper/Architecture-of-web-applications-based-on-Angular-A-Cincovi%C4%87-Del%C4%8Dev/5013f6f680a978ba985a731a90e2fbee21c3b99>
- CONCEITOS. (2014). *Software proprietário*. <https://conceitos.com/software-proprietario/>. São Paulo, Brasil. Acedido a 10 de julho de 2021.

- COSME, António. (2012). *Projeto em Sistemas de Informação Geográfica*. Lidel – edições técnicas, Lda., Lousã – Portugal, pág. 12 e pág. 238.
- COUTINHO, José Pereira. (2012). *Religião e outros conceitos*. Sociologia, Revista da Faculdade de Letras da Universidade do Porto, Vol. XXIV, pág. 171-193 – Porto, Portugal.
- CUNHA, Sara Maria Maia. (2009). *O SIG ao serviço do ordenamento do território: modelo de implementação. Trabalho de Projecto aplicado ao Município de Felgueiras*. (Dissertação de Mestrado – Universidade do Porto. Não Publicado). <https://repositorio-aberto.up.pt/bitstream/10216/20358/2/mestsaracunhasig000085086.pdf>. Acedido em 10 de julho de 2021.
- DIRECTIVA INSPIRE. (2007). *A infraestrutura da União Europeia de informação geográfica*. <https://eur-lex.europa.eu/legal-content/PT/LSU/?uri=celex:32007L0002>. Acedida a junho de 2022.
- DRUBI, Rodrigo Sanches. (s.d). *A Religião do Livro*. Paulinas. <https://www.paulinas.org.br/dialogo/pt-br/?system=paginas&action=read&id=5902>. Acedida a 13 de julho de 2021.
- ESRI PORTUGAL (2021). *O que são os SIG?*. <https://www.esri-portugal.pt/pt-pt/o-que-sao-os-sig/overview> acedido em 10 de julho de 2021.
- EUROSTAT. (2021). *European Statistics*. <https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/countries>. Acedido a 12 de julho de 2021.
- Federal Geography Data Committee (FGDC). (2010). *Introduction to Geospatial Web Services*. https://www.fgdc.gov/training/nsdi-training-program/materials/GeoWebservices_Intro_20100604.pdf. Acedido a 10 de julho de 2021.
- FONTE, Cidália. (2021). *Material de apoio a unidade curricular Modelação e Análise de Dados Geográficos, do curso de mestrado MTIG*. (Não publicado): Departamento de Matemática – FCTUC – Universidade de Coimbra.

FU, Pinde. (2018). *Getting to know Web GIS*. Third edition. Redlands, California : Esri Press.

GAARDER, Jostein, HELLERN, Victor, NOTAKER, Henry. (1952). *O livro das religiões*.

Tradução: Isa Mara Lando; revisão técnica e apêndice Antônio Flavio Pierucci. — São Paulo: Companhia das Letras.

GISGEOGRAPHY. (2021). *Commercial GIS Software: List of Proprietary Mapping Software*.

<https://gisgeography.com/commercial-gis-software/>. Acedido a 10 de julho de 2021.

GNU. (2021). *O que é o software livre?* [https://www.gnu.org/philosophy/free-sw.pt-](https://www.gnu.org/philosophy/free-sw.pt-br.html#mission-statement)

[br.html#mission-statement](https://www.gnu.org/philosophy/free-sw.pt-br.html#mission-statement)

GOURLEY, David, TOTTY, Brian, SAYER, Marjorie, REDDY, Sailu, AGGARWAL, Anshu.

(2002). *HTTP The definitive guide*. O'Reilly Media. Califórnia, EUA.

GUERRA, Marco. (2021). *No mundo 340 milhões de cristãos perseguidos. Covid agrava discriminações*.

Vatican News. <https://www.vaticannews.va/pt/igreja/news/2021-01/perseguiacao-cristaos-coronavirus-relatorio-2021.html>. Acedida a 14 de julho de 2021.

HARRINGTON, Jan L. (2016). *Relational Database Design and Implementation, Four edition*. Elsevier,

Massachusetts, EUA.

HURLBUT, Jesse Lyman. (1967). *The Story of the Christian Church*. Zondervan Publishing House.

Grand Rapids, Michigan, EUA.

INPE (2021). *Tutorial de Geoprocessamento*.

http://www.dpi.inpe.br/spring/portugues/tutorial/introducao_geo.html acedido em 10 de julho de 202.

ISRAEL. (2021). *Embaixada de Israel em Portugal*.

<https://embassies.gov.il/Lisboa/AboutIsrael/history/Pages/HISTORIA-Tempos-biblicos.aspx>. Acedida a 13 de julho de 2021.

- ISRIC. (2021). *World Soil Information*. <https://www.isric.org/web-coverage-services-wcs>. Acedido a 10 de julho de 2021.
- JOHNSON, Todd M, GRIM, Brian J. (2013). *Methodology of the World Christian Database*. Adaptação de *The World's Religions in Figures: An Introduction to International Religious Demography* (Oxford: Wiley---Blackwell, 2013).
- JOHNSON, Todd M, ZURLO, Gina A. (2021). *World Christian Database* (Leiden/Boston: Brill, 2021).
- JOLAYIA, Emmanuel. (2020). *Quick Introduction to GIS and WebGIS*. <https://jeafreezy.hashnode.dev/section-1-quick-introduction-to-gis-and-webgis-cki8iqqgh02a6hps15dgm5463>. Acedido a 10 de julho de 2021.
- LIU, Zhenpei. (2020). *Pipeline Spatial Data Modeling and Pipeline WebGIS. Digital Oil and Gas Pipeline: Research and Practice*. Springer Nature, Switzerland, pág.120-121.
- LUNA, Fernando, MILLAHUAL, Claudio Peña, IACONO, Matías. (2018). *PROGRAMACION WEB Full Stack 13 - PHP: Desarrollo frontend y backend*. RedUsers, Argentina.
- MASSÉ, Mark. (2012). *REST API Design Rulebook*. O'Reilly Media, Califórnia, EUA.
- MOZILLA. (s.d). *Métodos de requisição HTTP*. <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>. Acedida a junho de 2022.
- NUNES, Sérgio Sobral. (s.d.). *World Wide Web. Comunicações Digitais e Internet*. Universidade do Porto. <https://web.fe.up.pt/~ssn/disciplinas/cdi/www/5.html>. Acedido a 17 de julho de 2021.
- OGC. (s.d.). *Web Map Service*. <https://www.ogc.org/standards/wms>. Acedida a 17 de julho de 2021.
- OGC. (s.d.). *Web Feature Service*. <https://www.ogc.org/standards/wfs>. Acedida a 17 de julho de 2021.

- OPENSOFTE. (Junho, 2016). *Web service: o que é, como funciona, para que serve?* <https://www.opensoft.pt/web-service/>. Acedida a junho de 2022.
- OSI. (2021). *The Open Source Definition*. <https://opensource.org/docs/definition.php>. Acedido em 17 de julho de 2021.
- OSGeo. (2022). *What is Open Source?* <https://www.osgeo.org/about/what-is-open-source/>. Acedida a junho de 2022.
- ORACLE. (2022). *O que é um banco de dados relacional*. <https://www.oracle.com/br/database/what-is-a-relational-database/>. Acedido a 24 de janeiro de 2022.
- SAMPAIO, Lenita C. R. Ferraz. (2006). *Criacionismo e Evolucionismo*. Revista Faculdade de Ciências Médicas de Sorocaba – Vol. 8.
- SOFTWARE PUBLICO. (2021). *Lista de Software Open Source*. <https://www.softwarepublico.gov.pt/lista-de-software-open-source/>. Acedido a 10 de julho de 2021.
- TWF (2021). *The World FactBook*, <https://www.cia.gov/the-world-factbook/countries/world/#people-and-society>. Acedido a 17 de julho de 2021.
- VEIGA, Edison. (2021). *Paixão de Cristo: como foi a morte de Jesus, segundo a ciência*. BBC. Bled – Eslovênia. <https://www.bbc.com/portuguese/geral-56609774>. Acedida a 14 de julho de 2021.
- WORBOYS, Michael, DUCKHAM, Matt. (2004). *GIS A Computing Perspective*. Second edition, Flórida: CRC Press.

Anexo A: Mapa das doze tribos de Israel.

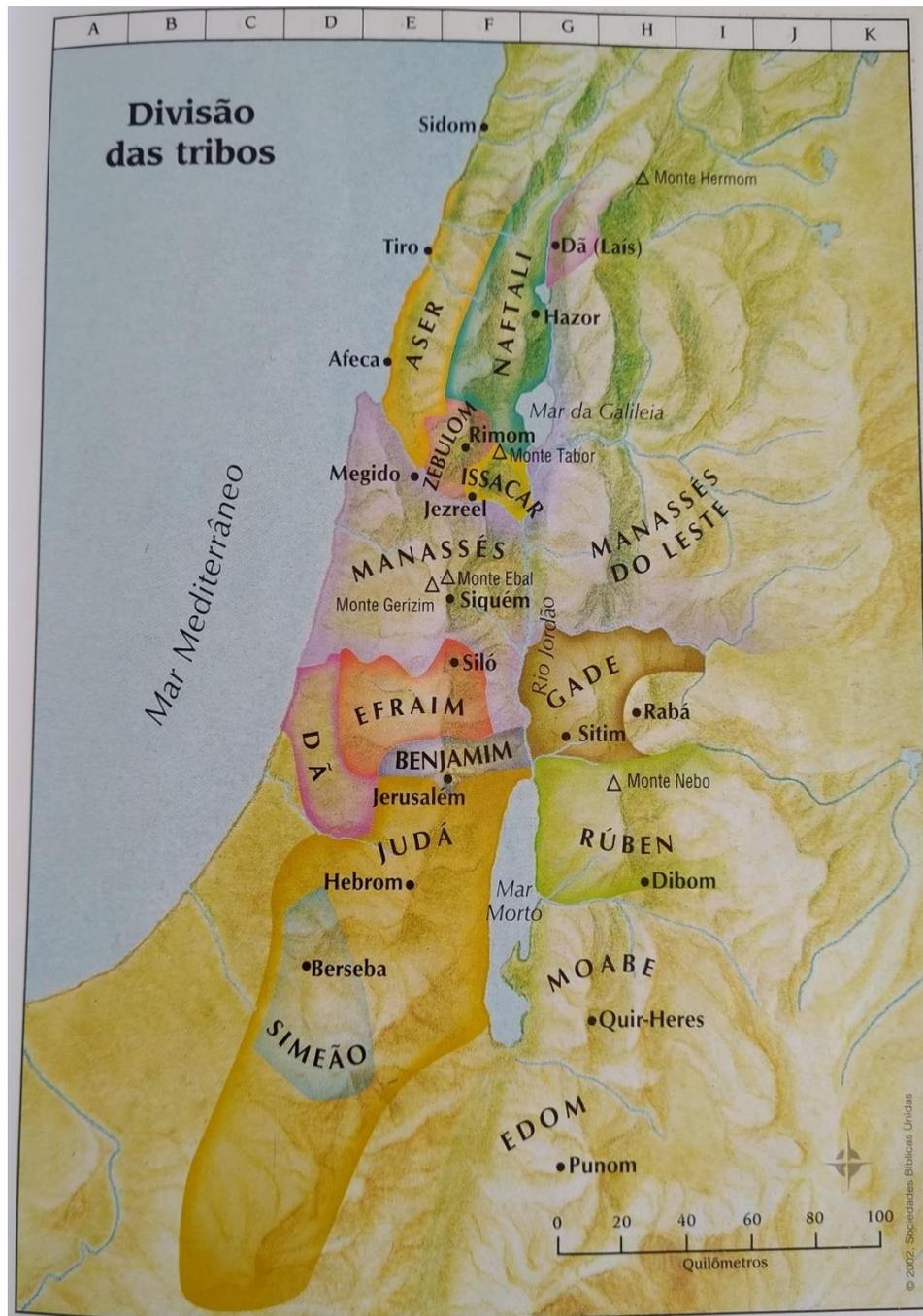


Figura 29: Divisão das doze tribos de Israel.
(Adaptação de: Bíblia Almeida Revista e Corrigida.)

Anexo B: Tabelas com as divisões dos tipos de territórios.

Tabela 11: Países que compõem as regiões do continente africano.

ÁFRICA				
África Austral	África Oriental	África Central	África Ocidental	África do Norte
Botswana	Burundi	Angola	Benin	Argélia
Essuatini	Comores	Camarões	Burkina Faso	Egito
Lesoto	Djibouti	República Centro-Africana	Cabo verde	Líbia
Namíbia	Eritreia	Chade	Costa do Marfim	Marrocos
África do Sul	Etiópia	Congo	Gâmbia	Sudão
	Quênia	RD do Congo	Gana	Tunísia
	Madagáscar	Guiné Equatorial	Guiné	Saara Ocidental
	Malawi	Gabão	Guiné-Bissau	
	Maurício	São Tomé e Príncipe	Libéria	
	Mayotte		Mali	
	Moçambique		Mauritânia	
	Reunião		Níger	
	Ruanda		Nigéria	
	Seychelles		Santa Helena	
	Somália		Senegal	
	Sudão do Sul		Serra Leoa	
	Tanzânia		Togo	
	Uganda			
	Zâmbia			
	Zimbábue			

Tabela 12: Países que compõem as regiões do continente americano.

AMÉRICA			
Caribe	América do Sul	América do Norte	América Central
Anguilla	Argentina	Bermudas	Belize
Antigua e Barbuda	Bolívia	Canadá	Costa Rica
Aruba	Brasil	Groenlândia	El Salvador
Bahamas	Chile	São Pedro e Miquelon	Guatemala
Barbados	Colômbia	Estados Unidos da América	Honduras
Ilhas Virgens Britânicas	Equador		México
Holanda caribenha	Ilhas Malvinas		Nicarágua
Ilhas Cayman	Guiana Francesa		Panamá
Cuba	Guiana		
Curaçao	Paraguai		
Dominica	Peru		
República Dominicana	Suriname		
Grenada	Uruguai		
Guadalupe	Venezuela		
Haiti			
Jamaica			
Martinica			
Montserrat			
Porto Rico			
São Cristóvão e Neves			
Santa Lúcia			
São Vicente			
São Martinho			
Trinidad e Tobago			
Ilhas de Turks e Caicos			
Ilhas Virgens Americanas			

Tabela 13: Países que compõem as regiões do continente asiático.

ÁSIA				
Sudeste asiático	Sul da Ásia	Ásia Central	Ásia Ocidental	Ásia Oriental
Brunei	Afganistão	Cazaquistão	Armênia	China
Camboja	Bangladesh	Quirguistão	Azerbaijão	Hong Kong
Indonésia	Butão	Tajiquistão	Bahrain	Japão
Laos	Índia	Turcomenistão	Chipre	Macau
Malásia	Irã	Uzbequistão	Georgia	Mongólia
Myanmar	Maldivas		Iraque	Coreia do Norte
Filipinas	Nepal		Israel	Coreia do Sul
Cingapura	Paquistão		Jordânia	Taiwan
Tailândia	Sri Lanka		Kuwait	
Timor-Leste			Líbano	
Vietnã			Omã	
			Palestina	
			Catar	
			Arábia Saudita	
			Síria	
			Turquia	
			Emirados Árabes Unidos	
			Iémen	

Tabela 14: Países que compõem as regiões do continente europeu.

EUROPA			
Europa Ocidental	Sul da Europa	Europa Oriental	Norte da Europa
Áustria	Albânia	Bielo-Rússia	Ilhas do Canal
Bélgica	Andorra	Bulgária	Dinamarca
França	Bósnia e Herzegovina	República Tcheca	Estônia
Alemanha	Croácia	Hungria	Ilhas Faroé
Liechtenstein	Gibraltar	Moldova	Finlândia
Luxemburgo	Grécia	Polônia	Islândia
Mônaco	Santa Sé	Romênia	Irlanda
Países Baixos	Itália	Rússia	Ilha de Man
Suíça	Kosovo	Eslováquia	Letônia
	Malta	Ucrânia	Lituânia
	Montenegro		Noruega
	Macedônia do Norte		Suécia
	Portugal		Reino Unido
	San Marino		
	Sérvia		
	Eslovênia		
	Espanha		

Tabela 15: Países que compõem as regiões do continente da Oceania.

OCEANIA			
Melanésia	Micronésia	Polinésia	Austrália-Nova Zelândia
Fiji	Guam	Samoa Americana	Austrália
Nova Caledônia	Kiribati	Ilhas Cook	Nova Zelândia
Papua Nova Guiné	Ilhas Marshall	Polinésia Francesa	
Ilhas Salomão	Micronésia	Niue	
Vanuatu	Nauru	Samoa	
	Ilhas Mariana do Norte	Tokelau	
	Palau	Tonga	
		Tuvalu	
		Wallis e Futuna Islands	

Anexo C: Apresentação das tabelas com os requisitos funcionais.

Tabela 16: Requisito funcional para a gestão dos utilizadores.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
07	<i>Admin User</i>	<i>Management Users</i>	GET	<i>Admin Page</i>	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador acessa a página de administração do Django Rest framework (DRF); 2 – Entra na parte dedicada a gestão de utilizadores; 3 – Lê a lista todos os utilizadores;	1 – O utilizador sem autenticação, não vai conseguir aceder a página de gestão do DRF.
08	<i>Admin User</i>	<i>Management Users</i>	POST	<i>Admin Page</i>	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador acessa a página de administração do DRF; 2 – Entra na parte dedicada a gestão de utilizadores; 3 – Adiciona um novo utilizador, preenchendo os campos necessários;	1 – O utilizador sem autenticação, não vai conseguir aceder a página de gestão do DRF.
09	<i>Admin User</i>	<i>Management Users</i>	PUT	<i>Admin Page</i>	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador acessa a página de administração do DRF; 2 – Entra na parte dedicada a gestão de utilizadores; 3 – Atualiza um utilizador, preenchendo os campos necessários;	1 – O utilizador sem autenticação, não vai conseguir aceder a página de gestão do DRF.
10	<i>Admin User</i>	<i>Management Users</i>	DELETE	<i>Admin Page</i>	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador acessa a página de administração do DRF; 2 – Entra na parte dedicada a gestão de utilizadores; 3 – Elimina o utilizador selecionado;	1 – O utilizador sem autenticação, não vai conseguir aceder a página de gestão do DRF.

Tabela 17: Requisito funcional para a gestão dos tipos de territórios.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
11	Admin User	Management Territory Type	GET	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os tipos de territórios da base de dados; 3 – Obtendo a lista através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
12	Admin User	Management Territory Type	POST	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar um ou vários novos registos na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
13	Admin User	Management Territory Type	PUT	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar um ou vários registos da base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
14	Admin User	Management Territory Type	DELETE	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar um ou vários registos da base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 18: Requisito funcional para a gestão dos anos.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
15	Admin User	Management Year	GET	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os anos da base de dados; 3 – Obtendo a lista através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
16	Admin User	Management Year	POST	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar um ou vários novos registos na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
17	Admin User	Management Year	PUT	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar um ou vários registos da base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
18	Admin User	Management Year	DELETE	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar um ou vários registos da base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 19: Requisito funcional para a gestão dos indicadores.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
19	Admin User	Management Indicators	GET	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os indicadores da base de dados; 3 – Obtendo a lista através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
20	Admin User	Management Indicators	POST	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
21	Admin User	Management Indicators	PUT	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
22	Admin User	Management Indicators	DELETE	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 20: Requisito funcional para a gestão dos territórios.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
23	Admin User	Management Territory	GET	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os territórios da base de dados; 3 – Obtendo a lista através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
24	Admin User	Management Territory	POST	DRF Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar o(s) registo(s) na base de dados; 3 – Podendo o fazer através do <i>package</i> LayerMapping do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
25	Admin User	Management Territory	PUT	DRF Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o(s) registo(s) na base de dados; 3 – Podendo o fazer através do <i>package</i> LayerMapping do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
26	Admin User	Management Territory	DELETE	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 21: Requisito funcional para a gestão das *layers*.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
27	Admin User	Management Layer	GET	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todas as <i>layers</i> da base de dados; 3 – Obtendo a lista através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
28	Admin User	Management Layer	POST	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
29	Admin User	Management Layer	PUT	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
30	Admin User	Management Layer	DELETE	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 22: Requisito funcional para a gestão da população.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
31	Admin User	Management Population	GET	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os da tabela população da base de dados; 3 – Obtendo a lista através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
32	Admin User	Management Population	POST	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
33	Admin User	Management Population	PUT	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
34	Admin User	Management Population	DELETE	Admin Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o(s) registo(s) na base de dados; 3 – Podendo o fazer através da página de gestão do DRF ou por <i>scripts</i> em Python.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no DRF, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 23: Requisito funcional para a gestão dos *Workspaces* do GeoServer.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
35	Admin User	Management GeoServer Workspace	GET	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os <i>Workspaces</i> do GeoServer; 3 – Obtendo a lista através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
36	Admin User	Management GeoServer Workspace	POST	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar um novo <i>Workspace</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
37	Admin User	Management GeoServer Workspace	PUT	Admin GeoServer Page	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o <i>Workspace</i> ; 3 – Podendo o fazer através da página do GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro.
38	Admin User	Management GeoServer Workspace	DELETE	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o <i>Workspace</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 24: Requisito funcional para a gestão dos *Stores* do GeoServer.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
39	Admin User	Management GeoServer Store	GET	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os <i>Stores</i> do GeoServer; 3 – Obtendo a lista através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
40	Admin User	Management GeoServer Store	POST	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar um novo <i>Store</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
41	Admin User	Management GeoServer Store	PUT	Admin GeoServer Page	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o <i>Store</i> ; 3 – Podendo o fazer através da página do GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro.
42	Admin User	Management GeoServer Store	DELETE	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o <i>Store</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 25: Requisito funcional para a gestão das *Layers* do GeoServer.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
43	<i>Admin User</i>	<i>Management GeoServer Layers</i>	GET	<i>Admin GeoServer Page</i> Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todas as <i>Layers</i> do GeoServer; 3 – Obtendo a lista através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
44	<i>Admin User</i>	<i>Management GeoServer Workspace</i>	POST	<i>Admin GeoServer Page</i> Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue adicionar uma nova <i>Layer</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
45	<i>Admin User</i>	<i>Management GeoServer Layers</i>	PUT	<i>Admin GeoServer Page</i>	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar a <i>Layer</i> ; 3 – Podendo o fazer através da página do GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro.
46	<i>Admin User</i>	<i>Management GeoServer Layers</i>	DELETE	<i>Admin GeoServer Page</i> Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar a <i>Layer</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Tabela 26: Requisito funcional para a gestão dos *Styles* do GeoServer.

RF	Ator	Nome	Método HTTP	Onde	Condição	Fluxo principal	Fluxo secundário
47	Admin User	Management GeoServer Styles	GET	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue listar todos os <i>Styles</i> do GeoServer; 3 – Obtendo a lista através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá aceder as informações. 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
48	Admin User	Management GeoServer Styles	POST	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue fazer o <i>upload</i> do ficheiro .sld criando um <i>Style</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá adicionar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.
49	Admin User	Management GeoServer Styles	PUT	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue atualizar o <i>Style</i> através de um novo ficheiro .sld; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá atualizar o(s) registo(s). 2 – Receberá uma mensagem de erro.
50	Admin User	Management GeoServer Styles	DELETE	Admin GeoServer Page Jupyter Notebook	Utilizador com as credenciais e permissões autenticadas pelo sistema.	1 – Utilizador insere as credenciais. 2 – Utilizador consegue eliminar o <i>Style</i> ; 3 – Podendo o fazer através da página do GeoServer ou por <i>scripts</i> em Python, utilizando o <i>package</i> geo.GeoServer.	1 – Quando não inserir o <i>username</i> e <i>password</i> corretamente no GeoServer, o sistema não permitirá eliminar o(s) registo(s). 2 – Receberá uma mensagem de erro. 3 – Do mesmo modo, quando tentar aceder através das linhas de código.

Anexo D: Linhas de código exemplificando a manipulação dos dados.

CHRISTIANS-API | Management of Territory Data

URL: /api/territorys/

Methods: GET | POST | PUT | DELETE

```
import requests
import json
import IPython.display

API_MAIN_URL = 'http://127.0.0.1:8000'

URL = f'{API_MAIN_URL}/api/territorys/'

HEADERS={
    'content-type' : 'application/json',
}
```

Method GET – Listing Territory data

```
r = requests.get(URL, headers=HEADERS)
IPython.display.JSON(r.json())
```

Method POST – To add Territory data

```
import geopandas as gp

refX = '/home/yurimtig/django/shp/world.shp'

name_en = 'NAME_EN'
name_pt = 'NAME_PT'
tipo = 'TYPE'

df = gp.read_file(refshp)

for i, row in df.iterrows():
    data = {

        'name_en' : row[name_en],
        'name_pt' : row[name_pt],
        'geom'    : row.geometry.wkt,
        'type'    : row[tipo],

    }
```

```
r = requests.post(URL, headers=HEADERS, data=json.dumps(data))
```

```
IPython.display.JSON(r.json())
```

Method PUT - Updating Territory data

```
a = 1
for i, row in df.iterrows():
    _url = f'{URL}{a}/'
    data = {

        'name_en' : row[name_en],
        'name_pt' : row[name_pt],
        'geom'     : row.geometry.wkt,
        'type'     : row[tipo],

    }

    r = requests.put(_url, headers=HEADERS, data=json.dumps(data))
    a = a+1

    if r.status_code != 201:
        break

    if i > 100:
        break
```

```
IPython.display.JSON(r.json())
```

Method DELETE – To Delete Territory data

```
r = requests.delete(URL'id/', headers=HEADERS)
```

```
IPython.display.JSON(r.json())
```

```
#For delete many territorys
```

```
for i in range(1,234):
    _url = f'{URL}{i}/'
    r = requests.delete(_url, headers=HEADERS)
```