



Visualização 3D interativa de informação meteorológica georreferenciada

DAMIEN GONÇALVES BARROS

outubro de 2021

Visualização 3D interativa de informação meteorológica georreferenciada

Damien Gonçalves Barros

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Filipe de Faria Pacheco

« A todos aqueles que me fizeram continuar »

Resumo

Feita uma análise às soluções existentes e disponíveis atualmente, no que toca a aplicações e mapas de visualização de informação meteorológica, foi notada a falta de uma aplicação em que esta mostre o estado do tempo em formato 3D. É nesse contexto que se insere este projeto.

No presente documento está então descrito o desenvolvimento de uma aplicação baseada num mapa, onde o utilizador pode escolher uma dada localização e verificar o estado do tempo nesse mesmo local. Ao selecionar a localização pretendida, o utilizador vai poder visualizar um ícone em formato 3D representativo do estado do tempo atual no exato local que foi pesquisado.

Para o desenvolvimento deste mesmo projeto, vão ser estudadas várias tecnologias que podem ser aplicadas, desde serviços de mapas até bibliotecas externas que podem fornecer os dados necessários á aplicação. Já as tecnologias base a serem usadas são o JavaScript, HTML e CSS.

Palavras-chave: Mapa, Meteorologia, 3D, Localização, Geocódigo, Web

Abstract

After an analysis to the available and existent solutions, in terms of applications and maps for weather data visualization, it was noticed the lack of an application that can show the description of the weather in an 3D format. This project is inserted in that context.

In this document it is described the development of an application based on a map where the user can choose the location e verify the weather description of that exact same location. By selecting the chosen location, the user will see an 3D icon representative of the real-time weather in the localization searched.

For the development of this project it will be studied some technologies that could be applied from map services to external libraries that could give the necessary data to the application. The base technologies to be used are JavaScript, HTML and CSS.

Keywords: Map, Weather, 3D, Localization, Geocoding, Web

Agradecimentos

Queria agradecer ao Professor Doutor Filipe de Faria Pacheco a total disponibilidade, orientação e aconselhamento no desenvolvimento de todo o projeto.

Ao Instituto Superior de Engenharia do Porto e ao Departamento de Engenharia Informática pela qualidade de ensino e por também me terem fornecido todos os conhecimentos adquiridos.

Á minha família e amigos pelo constante incentivo durante esta etapa.

Índice

1	Introdução	7
1.1	Contexto	7
1.2	Interpretação do Problema	7
1.3	Objetivo	8
1.4	Estrutura do Documento	8
2	Estado da Arte	10
2.1	Meteorologia em tempo real	10
2.2	API para dados climatéricos	10
2.2.1	Dark Sky	10
2.2.2	OpenWeatherMap.....	11
2.2.3	AccuWeather	11
2.2.4	ClimaCell.....	12
2.2.5	weatherstack.....	13
2.2.6	Quadro-resumo.....	13
2.3	API de Serviços de Mapas.....	14
2.3.1	Google Maps API	14
2.3.2	Leaflet.....	15
2.3.3	Mapbox GL JS	15
2.3.4	Quadro-resumo	16
2.4	Aplicações Genéricas	17
2.3.1	Dark Sky (iOS, web)	17
2.3.2	The Weather Channel (Android, iOS, web)	18
2.3.3	AccuWeather (Android, iOS, web).....	20
2.3.4	Weather Underground (Android, iOS, web).....	21
2.3.5	Google Weather (web).....	22
2.3.6	Weather/Tempo (iOS App).....	23
2.3.7	Portal IPMA (Android, iOS, web)	24
2.3.8	Foreca (Android, iOS, web)	25
2.5	Websites de informação meteorológica com foco na geografia.....	26
2.4.1	Windy	26
2.4.2	Foreca	26
2.4.3	The Weather Channel.....	28
2.6	Formas de apresentar meteorologia em canais de TV	29
3	Tecnologias	31
3.1	Tecnologias de base.....	31
3.1.1	HTML.....	31
3.1.2	CSS	31
3.1.3	JavaScript.....	32
3.1.4	Web Server for Chrome	32

3.2	Frameworks e bibliotecas.....	33
3.2.1	Three.js.....	33
3.2.2	Node.js.....	34
3.2.3	npm.....	35
3.2.4	Jest.....	35
3.3	Modeladores 3D.....	36
3.3.1	Blender 3D.....	36
4	Design	37
4.1	Objetivos.....	37
4.2	Requisitos.....	37
4.2.1	UC01 - Interagir com o Mapa.....	38
4.2.2	UC02 - Procurar Localização	39
4.2.3	UC03 - Ver meteorologia 3D.....	40
4.3	Limitações	40
4.4	Diagrama de Componentes	40
4.5	Soluções Alternativas	41
5	Implementação	43
5.1	Estrutura.....	43
5.2	Mapa e pesquisa de locais	44
5.3	API de meteorologia	46
5.4	Ícones 3D.....	48
5.5	Web Server	53
5.6	Interface Gráfica	54
6	Avaliação.....	57
6.1	Testes Unitários	57
6.2	Testes Funcionais.....	59
6.3	Testes de Usabilidade	59
7	Conclusões.....	64
7.1	Objetivos realizados	64
7.2	Limitações e trabalho futuro.....	64
7.3	Apreciação final	65
	Referências	66
	Anexo - Questionário.....	69

Lista de Figuras

Figura 1 - Google Maps API	14
Figura 2 - Biblioteca Leaflet	15
Figura 3 - Dark Sky (iOS)	17
Figura 4 - The Weather Channel (Android)	18
Figura 5 - AccuWeather (Android)	20
Figura 6 - Weather Underground (Android).....	21
Figura 7 - Google Weather	22
Figura 8 - Weather (iOS).....	23
Figura 9 - IPMA (Android).....	24
Figura 10 – Foreca (web)	25
Figura 11 - Windy.....	26
Figura 12 - Mapa de precipitação Foreca.....	27
Figura 13 - Mapa de temperatura The Weather Channel.....	28
Figura 14 - Meteorologia RTP	29
Figura 15 - The Weather Channel	30
Figura 16 - France 2.....	30
Figura 17 - Web Server for Chrome	33
Figura 18 - Uso da biblioteca three.js numa página web.....	34
Figura 19 – Jest	35
Figura 20 - Menu principal Blender	36
Figura 21 - Diagrama de Casos de Uso.....	38
Figura 22 - Diagrama de sequência UC01	39
Figura 23 - Diagrama de sequência UC02	39
Figura 24 - Diagrama de sequência UC03	40
Figura 25 - Diagrama de Componentes	41
Figura 26 – Estrutura.....	43
Figura 27 - Caixa de pesquisa	45
Figura 28 - <i>Token</i> de acesso <i>OpenWeather</i>	46
Figura 29 - Retorno <i>API OpenWeather</i>	47
Figura 30 - Probabilidade de precipitação (pop)	48
Figura 31 - Blender3D: Modelação de "broken clouds"	49
Figura 32 - Exportação para formato <i>glTF</i> no <i>Blender3D</i>	52
Figura 33 - Pasta raíz Web Server for Chrome	53
Figura 34 – Vista inicial mapa.....	54
Figura 35 - Vista inicial satélite	55
Figura 36 - Pesquisa pela cidade Paris	55
Figura 37 - Testes unitários efetuados com sucesso	58
Figura 38 - Formulário pergunta nr.1.....	60
Figura 39 - Formulário pergunta nr.2.....	60
Figura 40 - Formulário pergunta nr.3.....	61

Figura 41 - Formulário pergunta nr.4.....	61
Figura 42 - Formulário pergunta nr.5.....	62
Figura 43 - Formulário pergunta nr.6.....	62
Figura 44 - Formulário pergunta nr.7.....	63

Lista de Tabelas

Tabela 1 - APIs meteorológicas.....	13
Tabela 2 - Serviços de mapas	16
Tabela 3 - Ícones 3D para cada estado do tempo.....	49
Tabela 4 - Testes Funcionais.....	59

Lista de Código

Código 1 - Bibliotecas <i>Mapbox</i>	44
Código 2 - <i>Token</i> de acesso e objeto <i>Map</i>	44
Código 3 - <i>Geocoder Mapbox</i>	45
Código 4 - Chamada API do local pesquisado	47
Código 5 - Chamada API OpenWeather para precipitação	47
Código 6 - Método <i>meteo_conditions()</i>	52
Código 7 - Funções testes unitários	57
Código 8 - Bloco de testes unitários	58

Acrónimos

Lista de Acrónimos

2D	<i>Computação Gráfica Bidimensional</i>
3D	<i>Computação Gráfica Tridimensional</i>
API	<i>Application Programming Interface</i>
CORS	<i>Cross-Origin Resource Sharing</i>
CSS	<i>Cascading Style Sheets</i>
gITF	<i>Graphics Language Transmission Format</i>
HTML	<i>Hyper Text Markup Language</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IPMA	<i>Instituto Português do Mar e da Atmosfera</i>
ISEP	<i>Instituto Superior de Engenharia do Porto</i>
JSON	<i>JavaScript Object Notation</i>
MEI	<i>Mestrado em Engenharia Informática</i>
SO	<i>Sistema Operativo</i>
UV	<i>Radiação ultravioleta</i>
XML	<i>Extensible Markup Language</i>

1 Introdução

Neste capítulo será apresentado resumidamente o problema a ser interpretado nesta dissertação, que consiste na visualização 3D do estado do tempo num mapa, sendo possível também escolher a localização do mapa através de georreferenciação.

Assim sendo, neste capítulo será abordado de forma geral, o contexto do problema e a sua própria definição, isto é, em que consiste. Por fim, serão também apresentados os objetivos a atingir com esta solução.

1.1 Contexto

A previsão e visualização de informação meteorológica, como temperatura e clima, é algo quase imprescindível nos dias de hoje. Mas das aplicações já existentes, foi notada a falta de uma plataforma onde seja possível visualizar em 3D e em tempo-real essa mesma informação meteorológica. É nesse mesmo contexto que este projeto se baseia.

O conceito base deste trabalho é explorar esta área através da combinação de apresentação dinâmica de objetos 3D com informação geográfica e meteorológica. O sistema final pode ser útil quer para utilizadores individuais (e.g. planear uma viagem a pé na natureza), quer em ambientes mais públicos, interativos (e.g. quiosques multimédia) ou não (e.g. ecrã com informação multimédia) e serão exploradas várias alternativas nestes cenários.

Com base num serviço de mapas, uma API de meteorologia e exportação de ícones tridimensionais, a solução proposta neste projeto tem como objetivo a criação de um mapa interativo onde seja possível visualizar em 3D informação meteorológica georreferenciada.

1.2 Interpretação do Problema

Os mapas e gráficos de dados meteorológicos, na sua maioria, apresentam sempre informação em formato 2D e pouco detalhados visualmente do real estado do tempo numa zona específica. Nesse aspeto, não há proveito de potencialidades existentes no 3D que poderiam facilitar a interpretação dos dados referidos.

As soluções normalmente encontradas são aplicações meramente informativas, com gráficos ou ícones 2D em que é feita uma pesquisa por um local/cidade específicos. As soluções com mapas também apesar de já mostrarem mais alguma informação como nebulosidade, sensação térmica, direção do vento, entre outros, falha neste aspeto de mostrar informação

tridimensional e acrescentar imersividade á plataforma de consulta de meteorologia que está a ser usada.

Atualmente, já alguns canais de TV tentam apresentar informação meteorológica em 3D, mas mesmo assim ainda não tem um resultado verdadeiramente 3D em que também o utilizador possa interagir com o mapa.

1.3 Objetivo

A solução a ser implementada terá como base um sistema de visualização 3D baseado em web que permite a apresentação de objetos 3D de acordo com dados obtidos em tempo-real (ou quase-real) de serviços de informação. O utilizador ao abrir a aplicação, terá acesso a um mapa com o estado do tempo atual em formato 3D.

Este poderá pesquisar por um local e vai ser direcionado para esse mesmo local onde irá aparecer um ícone 3D por cima do mapa representativo do estado do tempo atual que se verifica no local pesquisado pelo utilizador. Este poderá também interagir com o mapa de várias formas podendo ampliar, diminuir, rodar e arrastar o mapa.

1.4 Estrutura do Documento

Este documento encontra-se dividido em sete capítulos. Neste primeiro capítulo, Introdução, resume de forma sucinta o problema e o contexto em que este se insere, assim como os objetivos esperados.

No capítulo 2 – Estado da Arte, é feito um estudo às plataformas já existentes de informação meteorológica de forma a ser percebido o que já existe e o que seria interessante vir a ser feito. É começado por ser feita uma abordagem a APIs de dados climatéricos e de serviços de mapas assim como uma análise às aplicações móveis, web, e forma de apresentar meteorologia presentes na televisão.

No capítulo 3 – Tecnologias, são abordadas todas as tecnologias que serão utilizadas no desenvolvimento da solução, desde linguagens de programação, bibliotecas externas a modeladores 3D.

O capítulo 4 – Design, é dedicado ao design da solução. Desde requisitos a diagramas, de forma a mostrar como a solução deverá funcionar e como irá ser implementada.

O capítulo 5 – Implementação, como o nome sugere, trata-se da descrição da implementação de todo o projeto. É feita uma descrição dos passos mais importantes no desenvolvimento do projeto assim como algumas decisões tomadas.

Já no capítulo 6 – Avaliação, são descritos os principais procedimentos para testes à aplicação. São também analisados os resultados obtidos.

Finalmente, no capítulo 7 – Conclusões, são apresentadas as conclusões tiradas do projeto assim como possível trabalho futuro.

2 Estado da Arte

Neste capítulo irá ser descrito o estado da arte das tecnologias de informação de dados meteorológicos em tempo real.

2.1 Meteorologia em tempo real

No estudo deste projeto irá ser usada uma API (*Application Programming Interface*) que vai servir de ponte entre uma das bases de dados de dados climatéricos em tempo real e o “*frontend*” da aplicação a desenvolver.

Uma API meteorológica para além de dispor dos dados às quais a aplicação vai aceder, deve também fornecer dados sobre:

Temperatura – Sensação térmica;

Humidade – Nível de humidade presente na atmosfera;

Velocidade do vento – Velocidade em km/h a que as rajadas de vento incidem;

Direção do vento – Ângulo em que este incide;

Precipitação – Probabilidade de precipitação;

Nebulosidade – Densidade de nuvens presentes;

Nevoeiro – Nuvem muito baixa cuja base encontra-se no solo e pode reduzir a visibilidade;

2.2 API para dados climatéricos

Existe uma grande variedade de APIs de meteorologia disponíveis no mercado. Baseado em vários *rankings* e listagens [1] [2] [3] das APIs deste género mais usadas, foram seleccionadas para este capítulo as mais interessantes a nível de funcionalidades.

2.2.1 Dark Sky

Dark Sky [4], inicialmente conhecida como *Forecast.io* é uma das mais populares aplicações de meteorologia. É atualmente bastante popular entre aplicações móveis e um dos mais precisos na recolha e fornecimento de informação.

Vantagens:

- Fonte precisa de informação meteorológica local;
- Os dados meteorológicos são frequentemente atualizados;
- É guardado um histórico de informação meteorológica;
- Suporte para 39 idiomas;

Desvantagens:

- Alertas meteorológicos só disponíveis em alguns países;
- A versão gratuita só permite 100 pedidos (API *requests*) por mês;

2.2.2 OpenWeatherMap

O OpenWeatherMap [5] oferece informação para diferentes linhas temporais (minutos, horas, dias), como temperatura atual, histórico ou previsões num dado período de dias. Os dados meteorológicos nesta API são recolhidos por serviços de transmissão meteorológica por todo o mundo e por mais de 40 mil estações meteorológicas.

A integração da App é feita através de JSON e XML e o seu preço varia desde gratuito até planos pagos com mais funcionalidades.

Vantagens:

- Pode ser utilizado para visualização de meteorologia em tempo-real e previsões ao usar dados de diferentes cidades numa área;
- Documentação disponível para casos de uso tendo como finalidade a agricultura;
- Extrato do histórico de dados com um custo justo de 10USD/cidade;

Desvantagens:

- A maioria da documentação encontra-se em estado beta, sendo assim importante o cuidado comercial a usar na aplicação;
- Na versão gratuita só são disponibilizados dados para o dia atual;

2.2.3 AccuWeather

AccuWeather [6] é um dos nomes mais conhecidos no que toca a meteorologia. Oferece informação atual, histórico de previsões e também informação relativa a meteorologia em diversos pontos do mundo. Curiosamente, também oferece informação específica como, por

exemplo, a probabilidade da existência de mosquitos num certo local mediante o estado atmosférico.

A integração é feita através de chamadas *RESTfull* e a resposta formatada em JSON e JSONP. A versão gratuita está limitada a 50 chamadas por dia.

Vantagens:

- API para imagens de radar e satélite também com funcionalidades de localização;
- Funcionalidade de alertas com avisos de clima severo fornecido pelo governo e agências de meteorologia (tornando-o assim confiável);
- A API de alarmes de clima é interessante no sentido de permitir 1,5,10 ou 15 dias de alertas baseado em limites específicos;

Desvantagens:

- A funcionalidade de “*autocomplete*” tem um melhor funcionamento em idiomas latinos;

2.2.4 ClimaCell

A API ClimaCell [7] oferece *endpoints* com diversos tipos de campos de informação como clima, qualidade do ar, pólen encontrado no ar, risco nas estradas, probabilidade de incêndio, entre outros. Para além de dados em tempo real, possui também um histórico de previsões e previsões futuras.

A sua integração é compatível com AWS, AUTODESK, REST API com JSON e também suporte HTTPS.

Vantagens:

- Vasta biblioteca de funcionalidades;
- Possui uma boa interface;
- Design da API flexível, o que se torna num ponto a favor para vários tipos de aplicações;
- Plano gratuito bastante completo;

Desvantagens;

- Alguns aspetos encontram-se com erros;
- Todos os planos quer estes sejam gratuitos ou pagos têm chamadas á API limitadas;
- É necessário o uso de *backlinks* em todos os planos;

2.2.5 weatherstack

A weatherstack [8] é uma API mais direcionada para *websites* e aplicações móveis mais no contexto de utilização de *widgets* com disponibilização de informação meteorológica em tempo real. Para além de informação em tempo-real, oferece também um histórico de previsões assim com informação mundial.

Suporta chamadas REST API com respostas em JSON e JSONP. O HTTPS é apenas suportado em versões pagas.

Vantagens:

- API rápida e fácil de usar;
- Informação meteorológica precisa;
- API com capacidade para mais de mil milhões de chamadas por dia;
- Documentação completa e de fácil leitura;

Desvantagens:

- Plano gratuito bastante limitado;
- Não inclui localização no plano gratuito;

2.2.6 Quadro-resumo

Tabela 1 - APIs meteorológicas

	Dark Sky	OpenWeatherMap	AccuWeather	ClimaCell	weatherstack
Preço	Gratuito/Pago	Gratuito/Pago	Gratuito/Pago	Gratuito/Pago	Gratuito/Pago
Integração	JSON	JSON, XML	JSON, JSONP	AWS, Autodesk, JSON	JSON
Tempo-real	Não	Sim	Sim	Sim	Sim
Previsões	Sim	Sim	Sim	Sim	Sim (Pago)
Histórico	Sim	Sim	Sim	Sim	Sim (Pago)
Nr chamadas API (na versão gratuita)	1000/dia	60/dia	50/dia	1000/dia	1000/dia
Mapa	Não	Sim	Não	Sim	Não

Após uma análise detalhada e atendendo à Tabela 1, foi concluído que as API que parece mais adequada ao uso neste projeto são a OpenWeatherMap. Esta à partida, das APIs em análise, é a que dispõem de mais funcionalidades para o tipo de projeto em estudo assim como o facto dessas mesmas funcionalidades poderem ser obtidas de forma gratuita. Para além dos aspetos referidos, também possui uma boa integração com serviços de mapas.

2.3 API de Serviços de Mapas

2.3.1 Google Maps API

API própria do conhecido serviço *Google Maps* [9], representado na figura 1, esta apresenta bastantes funcionalidades como apresentação de mapas, rotas, lugares, elevação territorial, fusos horários, geolocalização, entre outros.

Tem 99% de cobertura no mundo [10]e para além de mapas é também possível usar rotas e lugares.



```
function initMap(): void {
  const directionsRenderer = new google.maps.DirectionsRenderer();
  const directionsService = new google.maps.DirectionsService();
  const map = new google.maps.Map(
    document.getElementById("map") as HTMLElement,
    {
      zoom: 14,
      center: { lat: 37.77, lng: -122.447 },
    }
  );
  directionsRenderer.setMap(map);

  calculateAndDisplayRoute(directionsService, directionsRenderer);
  (document.getElementById("mode") as HTMLInputElement).addEventListener(
    "change",
    () => {
      calculateAndDisplayRoute(directionsService, directionsRenderer);
    }
  );
}
```

Figura 1 - Google Maps API

Este apesar de ser um reconhecido serviço de mapas tem o problema de ter certas limitações como o facto de apenas ser gratuito temporariamente. É também necessário criar uma conta de desenvolvedor para usar esta API.

2.3.2 Leaflet

Leaflet [11], representado na figura 2, é uma biblioteca *JavaScript* para mapas interativos para *mobile* e também *desktop*. É uma aplicação de código-aberto, isto é, que qualquer pessoa pode usar sem custos adicionais. Tem também suporte para HTML5 e CSS3.

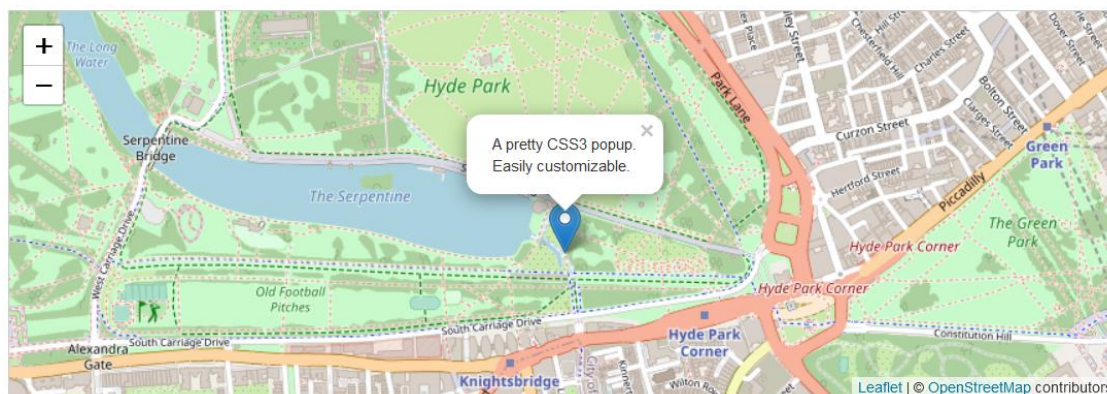


Figura 2 - Biblioteca Leaflet

Algumas das suas funcionalidades são:

- Camadas com vários estilos, como marcadores, imagens, etc.;
- *Scroll*, *zoom*, arrasto com o rato, navegação com o teclado;
- Suporte para vários *browsers*;
- Uso de funcionalidades presentes no CSS 3;
- Várias animações em *popups* e camadas do mapa;

2.3.3 Mapbox GL JS

É uma biblioteca *JavaScript* [12] para a construção de mapas e aplicações web. Pode ser usado num browser e/ou cliente, adicionadas interfaces com o utilizador e ser feita customização do mapa (por exemplo, ao escolher o estilo do mapa).

Alguns usos do Mapbox GL JS incluem:

- Visualização e animação de informação obtida geograficamente;
- Filtragem de funcionalidades no mapa;
- Uso de informação nas camadas do mapa;
- Amostras dinâmicas e personalização de informação do lado do cliente no mapa;

- Visualização de informação em 3D e animações;
- Adição de marcadores e *popups* no mapa de forma programada;

2.3.4 Quadro-resumo

Tabela 2 - Serviços de mapas

	Google Maps	Mapbox	Leaflet
Mapas próprios	Sim	Sim	Não
Suporte objetos 3D	Não	Sim	Não
Versão gratuita ilimitado	Não	Sim	Sim
Vários estilos	Sim	Sim	Sim
Visualização web 3D	Não	Sim	Sim

Com este projeto tendo como base um mapa onde é demonstrada informação meteorológica, foi necessário definir logo no início qual a API de mapas mais adequada. Inicialmente foi decidido o uso do Google Maps API visto este ser um dos mais conhecidos.

Após alguns testes e desenvolvimentos, houve um primeiro problema: O uso gratuito da API do Google Maps estava limitado a um certo período de tempo e não era possível rodar e inclinar o mapa na versão web. Neste sentido, foi necessário encontrar uma alternativa para a API de mapas e foi então escolhido o Mapbox GL JS. Este já não possuía as limitações presentes no Google Maps API e também apresenta bastantes funcionalidades interessantes para o uso neste projeto como por exemplo a integração com a biblioteca JavaScript 3D three.js.

O leaflet por sua vez, acabou por ser descartado, pois este na verdade não possui um serviço próprio de mapas. O que faria com que fosse necessário usar o Google Maps, Mapbox ou outro como mapa base para usar o leaflet. Desta forma, e atendendo á Tabela 2, o Mapbox GL JS é a melhor opção pois tendo já um serviço de mapas próprio, tem um vasto leque de funcionalidades necessárias ao projeto de forma gratuita.

2.4 Aplicações Genéricas

Nesta secção irão ser apresentadas algumas das aplicações de meteorologia mais conhecidas e listadas em *rankings* internacionais [13] de melhores aplicações deste segmento. Nos primeiros quatro subtópicos vão ser então descritas algumas dessas aplicações. Nos pontos seguintes vão ser descritas algumas aplicações que, embora não estejam necessariamente listadas nalgum tipo de ranking, são aplicações que não deixam de ter algumas funcionalidades interessantes e são usadas por um largo segmento de utilizadores, dada a sua simplicidade e a forma como estão inseridas tanto em sistemas operativos móveis e até em televisão.

2.3.1 Dark Sky (iOS, web)



Figura 3 - Dark Sky (iOS)

Dark Sky, Figura 3, considerada uma das melhores aplicações meteorológicas [14], esta que também já esteve disponível para o sistema operativo móvel Android, agora é exclusiva iOS (pois foi adquirida pela Apple) [15].

Com previsões de tempo precisos ao minuto (no que toca à precipitação), esta é uma aplicação bastante completa, onde cobre no geral todos os aspetos informativos a previsões meteorológicas. A aplicação é capaz também de fornecer informação sobre o vento, humidade, ponto de condensação, radiação UV, visibilidade e pressão atmosférica.

Possui também um mapa onde é possível ver informação sobre a temperatura num local específico (é possível pesquisar por rua), sensação térmica, nebulosidade, direção do vento, entre muitos outros, até cobertura de ozono. Curiosidade que os *developers*¹ deixaram no mapa: cobertura do mapa através de *emojis*. Onde tem uma cara sorridente onde o clima é favorável, e uma cara triste caso contrário.

A aplicação é também bastante personalizável, onde é possível escolher se a visualização de dados quer em graus Fahrenheit, Celsius, milhas, km/h, etc. Também é possível escolher o idioma.

Na aplicação para iOS possui também uma série de notificações onde o utilizador as pode ativar e receber alertas.

2.3.2 The Weather Channel (Android, iOS, web)

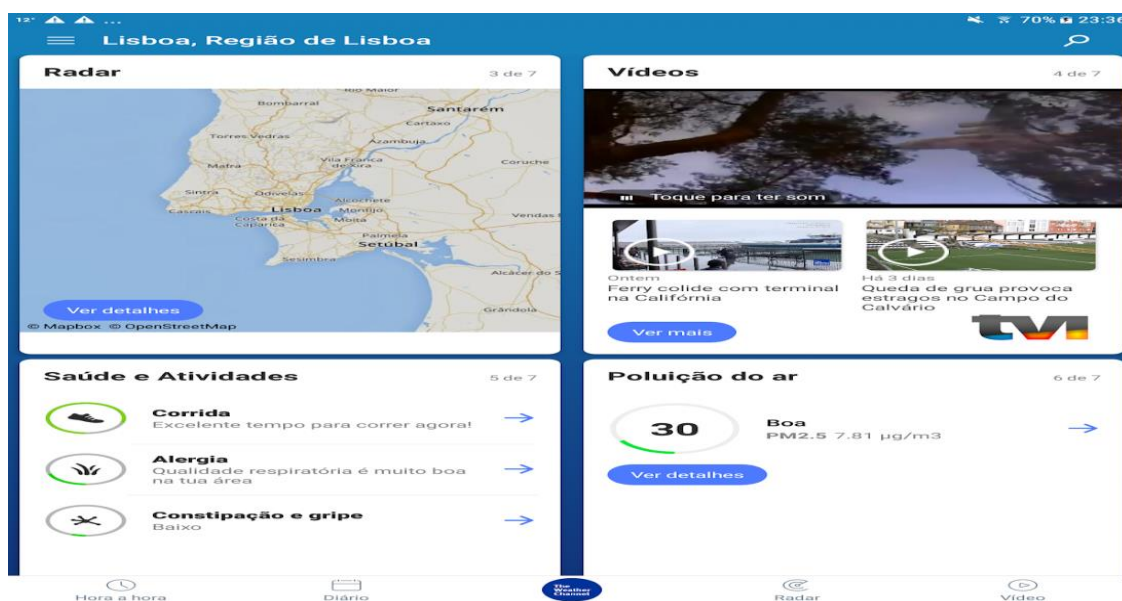


Figura 4 - The Weather Channel (Android)

¹ Developer/Software Developer – Desenvolvedor de software

The Weather Channel, Figura 4, pertencente à *IBM*, é também um dos portais meteorológicos mais conhecidos e utilizados. É uma plataforma bastante completa no que diz respeito a toda a informação considerada importante numa plataforma que tem como objetivo informação do estado do tempo.

É possível escolher a localização de onde se pretende consultar a meteorologia, embora ao usar o *website*, este faça a deteção automaticamente da localização do utilizador através do serviço de internet/rede móvel. Na plataforma é possível fazer a previsão diariamente, por horas, num período de 10 dias, apenas fins de semana e mensalmente. Alguns dados apresentados:

- Temperaturas mínimas e máximas
- Humidade
- Pressão
- Visibilidade
- Vento
- Ponto de condensação
- Índice de raios UV
- Fase da lua

Este também tem um mapa/radar em que é possível ver diferentes dados como concentração de nuvens e informação de temperatura através de cores.

Para além da plataforma web, esta também possui aplicações próprias para Android, iOS e Windows 10.

2.3.3 AccuWeather (Android, iOS, web)



Figura 5 - AccuWeather (Android)

A AccuWeather [21], Figura 5, embora que não tão completa como as anteriores, possui ainda assim várias funcionalidades interessantes, colocando-a assim também na lista das mais conhecidas e utilizadas. Possui aplicações para as plataformas Android, iOS e web.

Depois da pesquisa de um local é possível ver informação para este mesmo a cada hora, diariamente e mensalmente. Este, tal como o Dark Sky, também possui previsão de precipitação com intervalos de minutos.

Ao pesquisar informação mais detalhada, é possível verificar qualidade do ar, coordenadas e velocidade do vento, humidade, ponto de condensação, pressão atmosférica, nebulosidade, visibilidade e altitude das nuvens.

Quanto à consulta por mapas, embora não seja muito detalhado, para além da consulta de temperatura, é possível verificar tempestades ativas e por que ponto do globo estão a passar.

2.3.4 Weather Underground (Android, iOS, web)

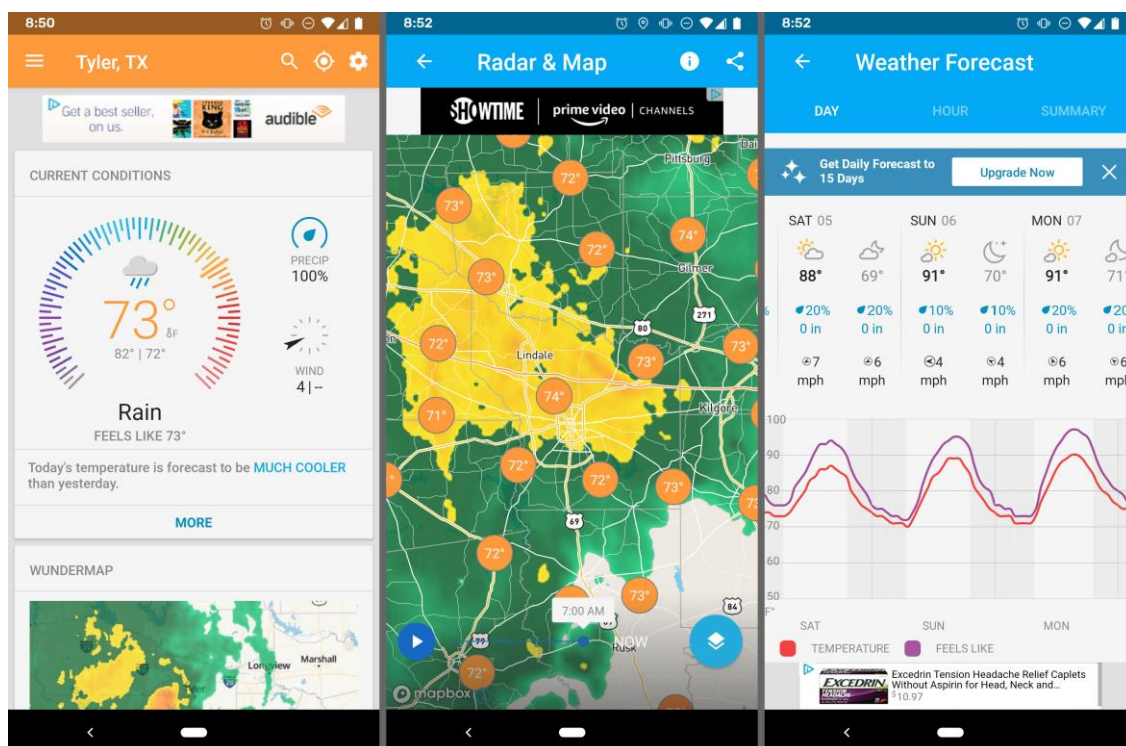


Figura 6 - Weather Underground (Android)

O Weather Underground, Figura 6, é também uma plataforma bastante conhecida (maioritariamente no mercado americano, dado as unidades de medida se encontrarem no formato americano, como por exemplo, graus Fahrenheit e milhas), possui também aplicações móveis para os sistemas operativos móveis mais conhecidos, assim como portal web. Ambas as aplicações móveis são gratuitas.

Esta plataforma permite também a procura de um local específico e é possível verificar informação do dia, por horas, num intervalo de 10 dias, por calendário (onde é escolhido mês e ano) e é também possível verificar um histórico de n dias. Para além das temperaturas mínimas e máximas é também possível verificar:

- Pressão atmosférica;
- Visibilidade;
- Nebulosidade;
- Ponto de condensação;
- Humidade;
- Precipitação;
- Profundidade de neve;
- Polén contido no ar;
- Qualidade do ar;

- Índice UV;

Quanto aos mapas deste, é apenas possível ver o mapa em formato radar ou satélite. De destacar também que é possível consultar alguma informação a nível de astronomia, como duração do tempo de luz visível, duração do tempo do dia e noite, crepúsculos civil, náutico e astronómico e também qual fase da lua irá surgir num dia específico.

2.3.5 Google Weather (web)

Meteorologia oferecida pelo portal Google. Independentemente de onde quer que o utilizador esteja ao escrever “tempo” no motor de busca Google este irá disponibilizar o estado do tempo no local em que o utilizador se encontra.



Figura 7 - Google Weather

É mostrada a temperatura atual, assim como as temperaturas mínimas e máximas para a semana seguinte. Esta temperatura pode ser mostrada em graus Celsius ou Fahrenheit.

Para além da temperatura, é também mostrado um pequeno ícone relativo ao clima e é possível também verificar a precipitação e vento, como representado na Figura 7.

2.3.6 Weather/Tempo (iOS App)



Figura 8 - Weather (iOS)

Aplicação nativa do sistema operativo para dispositivos móveis da Apple (Figura 8). Também é capaz de detetar a temperatura atual com base na localização. Adicionalmente, é também possível escolher e registar outras localizações, para o utilizador ter acesso e assim verificar a meteorologia sempre que pretender.

Este apresenta também uma previsão de dias das temperaturas mínimas e máximas assim como probabilidade de precipitação. Outro ponto interessante na aplicação é o fundo dinâmico em que, consoante esteja sol, chuva, ou outro, esse fundo pode mudar desde um sol radiante a uma tempestade intensa.

2.3.7 Portal IPMA (Android, iOS, web)



Figura 9 - IPMA (Android)

Portal do Instituto Português do Mar e da Atmosfera, Figura 9. Neste portal é possível ver as previsões meteorológicas para qualquer região em Portugal. Este também apresenta um pequeno ícone do clima atual. Para além desses aspetos são também apresentados para cada um desses dias:

- Temperaturas mínimas e máximas;
- Direção do vento;
- Probabilidade de precipitação;
- Sensação térmica;
- Avisos meteorológicos;
- Radiação UV;

2.3.8 Foreca (Android, iOS, web)

Foreca [16], uma das plataformas mais precisas² e a maior do seu tipo nos países nórdicos, é também uma das plataformas com uma análise de meteorologia bastante completa.



Figura 10 – Foreca (web)

Este possui previsões de hora em hora e de até 15 dias, como representado na Figura 10. Estas previsões são bastante detalhadas com temperaturas mínimas e máximas, humidade relativa, ponto de condensação, hora de nascer e pôr-do-sol, entre outros.

Possui também um mapa onde é possível ver graficamente informação relativa a:

- Precipitação;
- Temperatura;
- Nuvens;
- Satélite;
- Pressão atmosférica;
- Radiação UV;

² <https://www.smartertravel.com/most-accurate-weather-app/>

2.5 Websites de informação meteorológica com foco na geografia

Para além de *websites* que mostram dados meteorológicos meramente informativos, existem também uma variedade destes que para além dessa informação, mostram-na também de forma a ter a geografia como foco. Exemplo desses é o Windy (Figura 11).

2.4.1 Windy

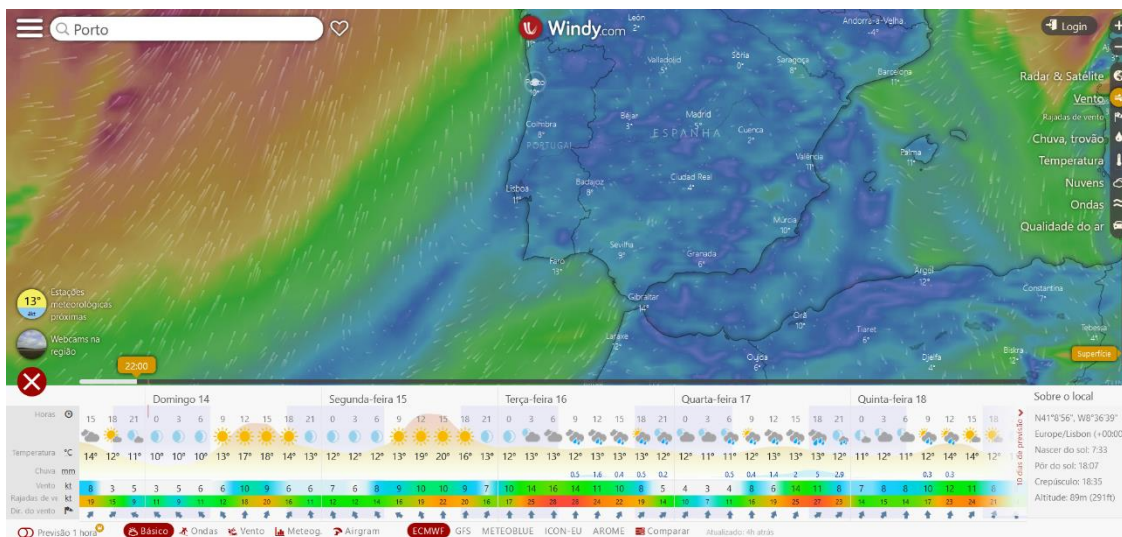


Figura 11 - Windy

O Windy [16] é um *website* de informação meteorológica essencialmente focado na geografia. Este possui informações em tempo real sobre vento, precipitação, nebulosidade, ondas, qualidade do ar, entre outros.

Para além do Windy, plataformas como o Foreca ou o The Weather Channel, mencionados acima também usam este tipo de infografia.

2.4.2 Foreca

O Foreca [17] também possui dados meteorológicos através de informação geográfica bastante detalhados.

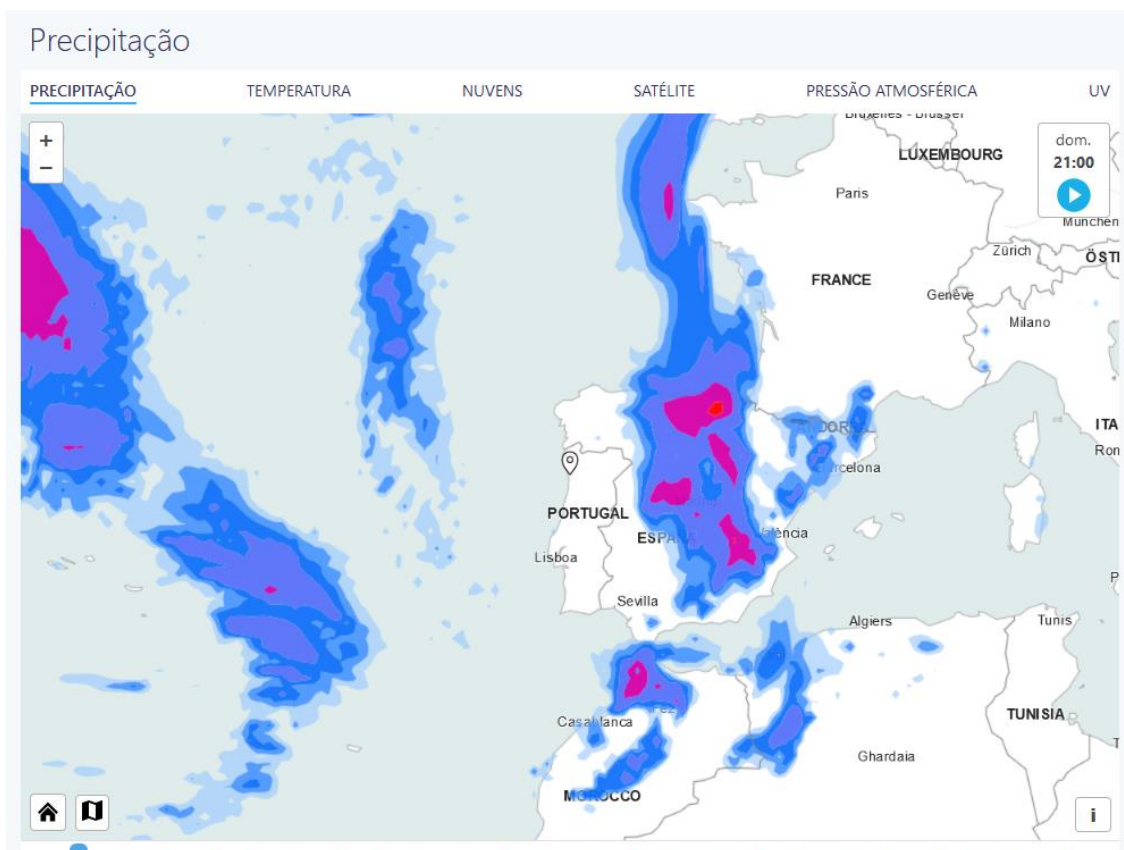


Figura 12 - Mapa de precipitação Foreca

Como é possível verificar na Figura 12, é possível consultar informação sobre precipitação, temperatura, nebulosidade, vista de satélite, pressão atmosférica e raios ultravioleta também através do Foreca.

2.6 Formas de apresentar meteorologia em canais de TV

Ainda antes de ser possível a visualização de meteorologia graficamente num *website*, a meteorologia já era apresentada, diariamente, por exemplo, nas notícias transmitidas na televisão. No caso da TV portuguesa, esta normalmente é representada por um mapa com valores de temperaturas máximas e mínimas sobre cada distrito do país.



Figura 14 - Meteorologia RTP

Na generalidade dos canais televisivos, a informação do estado do tempo é fornecida pelo Instituto Português do Mar e da Atmosfera, como mostra a Figura 14.

No que diz respeito a televisão internacional, The Weather Channel costuma optar por uma abordagem bastante interessante. Através do uso de realidade aumentada conseguem resultados bastantes bons a nível visual e conseguem tornar os gráficos mais interativos. Aquando do Furacão Florence, nos Estados Unidos da América, The Weather Channel conseguiu simular graficamente e através de AR (Augmented Reality – Realidade Aumentada), a altura que o nível da água poderia atingir (Figura 15).



Figura 15 - The Weather Channel

De outra forma, também o canal televisivo francês France 2, Figura 16, opta por mostrar um mapa com um esquema de cores representativas das temperaturas e sensação térmica.



Figura 16 - France 2

3 Tecnologias

Neste capítulo são apresentadas todas as tecnologias consideradas para a implementação do projeto final.

3.1 Tecnologias de base

3.1.1 HTML

Acrónimo de *HyperText Markup Language*, descreve a estrutura de uma página web. Consiste numa série de elementos/*tags* onde o texto a tratar é encapsulado. São estes mesmo elementos que dizem ao *browser*³ o tipo de conteúdo a mostrar.

É standard para qualquer página web desde a mais simples á mais complexa. Todos os ficheiros *html* começam com a *tag* `<!DOCTYPE html>` que define que se trata de um documento html e também ser entendido pelo *browser*. Este é seguido pela *tag* `<head>`, que possui metadados⁴ e `<title>` - título da página web que vai aparecer no separador do *browser*. Em termos estruturais, é finalizado pelo `<body>`, onde vai estar todo o conteúdo que vai efetivamente ser mostrado na página *web* a ser desenvolvida.

3.1.2 CSS

Acrónimo de *Cascading Style Sheets*, serve para personalizar o *HTML*, dando cor, diferentes fontes, tamanhos entre outros vários tipos de formatação e personalização.

³ Browser/Navegador – Permite navegar pela internet

⁴ Dados sobre outros dados (informação)

É normalmente representado por um ficheiro externo do *html* e possui terminologia *.css* que depois é chamado no `<head>` do *html* através do `href`. Um ficheiro CSS é constituído por seletores em que dentro destes encontram-se os parâmetros e respetivos valores a personalizar, exemplo: `h1 {color: blue}`. Desta forma todos os elementos `h1` presentes no *html* em que o ficheiro *css* está associado vão ter a cor azul.

Também pode ser usado dentro do próprio ficheiro *html*. Dentro da *tag* `<head>` do *html*, basta definir os estilos ao usar as *tags* `<style></style>`.

3.1.3 JavaScript

JavaScript é uma linguagem de programação específica para web onde permite implementar funcionalidades mais complexas. É uma linguagem orientada a objetos.

Este para ser associado a um ficheiro *html* tem que estar entre as *tags* `<script>`. As instruções JavaScript podem estar diretamente presentes dentro das *tags* `<head>` ou `<body>` como também ser usado um ficheiro JavaScript externo.

É este que vai manipular todo o comportamento de uma página *web*, desde filtragem de dados, registos, entre muitos outros. Tal como o *css*, este também pode ser definido/construído dentro do próprio ficheiro *html*, se colocado dentro das *tags* `<script>`.

3.1.4 Web Server for Chrome

É uma extensão [19] existente para Google Chrome em que funciona como servidor local (Figura 17). Este é necessário para “lançar” a aplicação web principalmente com recurso a ficheiros, como imagens, por exemplo, de várias fontes. Pode também ser usada como uma aplicação independente, isto é, é criado um atalho da aplicação no próprio computador sem que seja necessário o Google Chrome⁵ esteja ligado.

Para evitar a necessidade de se programar manualmente um servidor web, existe este plugin na loja do Google Chrome que o faz automaticamente e que se torna bastante útil para aplicações internas e pré-testes. Este mesmo plugin tem a particularidade de:

- Poder ser escolhida a pasta raiz do projeto a ser colocado no servidor;
- Colocar a extensão a correr em “background”;
- Ser acessível por rede local como para fora (abrindo uma porta no router);
- Prevenir o computador onde a aplicação está a correr de ficar suspenso;

⁵ Navegador web

- Pode também usar https;

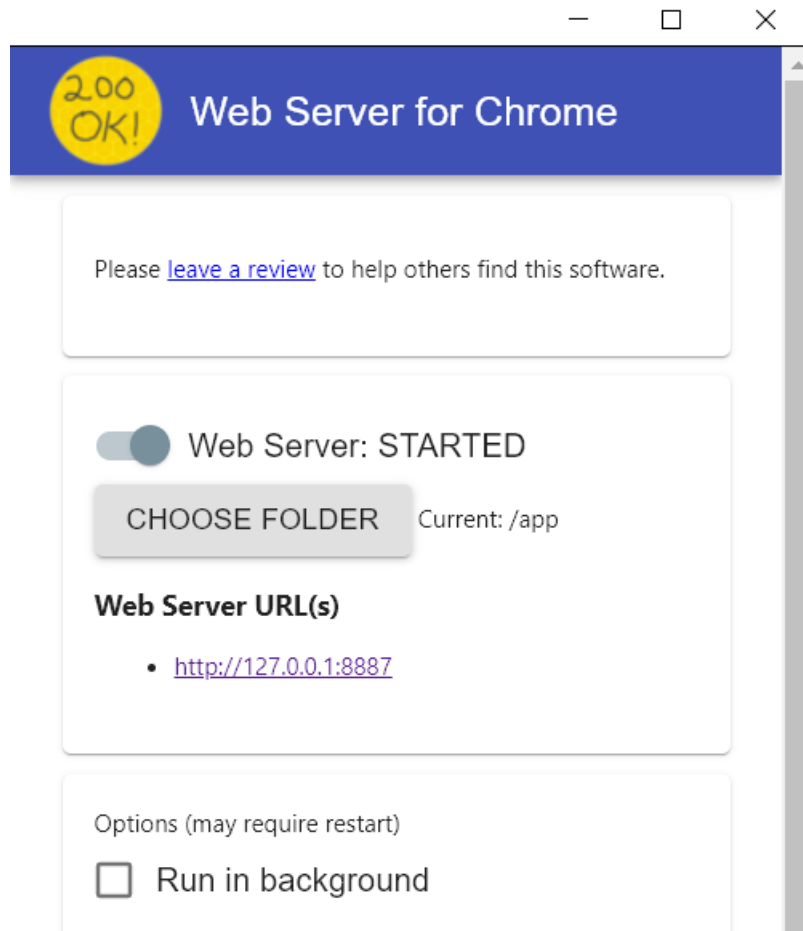


Figura 17 - Web Server for Chrome

3.2 Frameworks e bibliotecas

3.2.1 Three.js

Three.js [20] é uma biblioteca *JavaScript* usada para gráficos 3D na web. É essencialmente usado para acrescentar conteúdo 3D em páginas web.

Esta biblioteca pode servir para apresentar conteúdo estático ou animado. Em certas aplicações é possível animar e configurar o objeto 3D presente na página (Figura 18). Por exemplo, numa aplicação com uma cortina de pano seria possível mudar a direção e velocidade do vento e o objeto 3D reagiria consoante esses parâmetros.



Figura 18 - Uso da biblioteca three.js numa página web

3.2.2 Node.js

Node.js [21] é uma plataforma onde é possível compilar aplicações web. É “*open-source*”, isto é, uma aplicação de código-aberto e por isso gratuita. Pode ser executado em várias plataformas como Windows, Mac, Linux e usa JavaScript.

Por sua vez os ficheiros *node* são então ficheiros JavaScript que contêm tarefas que serão executadas em certos eventos, como por exemplo, um utilizador a tentar aceder a uma porta no servidor. Estes ficheiros têm que ser inicializados antes de surtirem algum efeito.

Alguns usos que o Node.js possui:

- Pode gerar conteúdo dinâmico numa página;
- Pode criar, abrir, ler, escrever, fechar e apagar ficheiros do servidor;
- Pode recolher dados de formulários;
- Pode modificar a base de dados o projeto alojado no servidor;

3.2.3 npm

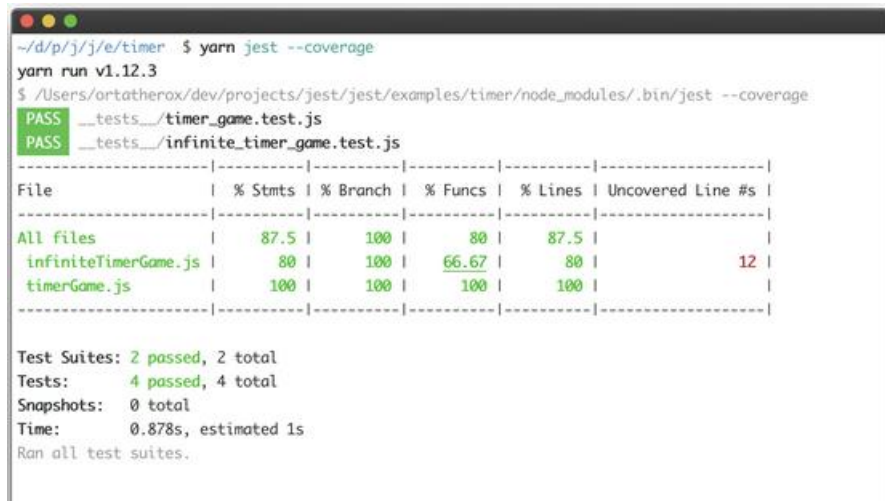
O npm (*Node Package Manager*) [22], é um sistema de gestão de pacotes. Consiste em três principais componentes:

- O seu próprio website – para pesquisa de pacotes existentes, criação de perfis, etc;
- CLI (*Command Line Interface*) – Corre num terminal onde é feita a maior parte da interação com o npm;
- Registo – Base de dados online com pacotes JavaScript públicos e privados;

Estes pacotes acima referidos são na generalidade *frameworks* e bibliotecas *JavaScript* que podem ser usados para as mais variadas aplicações.

3.2.4 Jest

Jest [23] é uma *framework* de testes *JavaScript*. É também instalado através do npm.



```
~/d/p/j/j/e/timer $ yarn jest --coverage
yarn run v1.12.3
$ /Users/ortatherox/dev/projects/jest/jest/examples/timer/node_modules/.bin/jest --coverage
PASS  __tests__/timer_game.test.js
PASS  __tests__/infinite_timer_game.test.js

File | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s |
-----|-----|-----|-----|-----|-----|
All files | 87.5 | 100 | 80 | 87.5 | |
infiniteTimerGame.js | 80 | 100 | 66.67 | 80 | 12 |
timerGame.js | 100 | 100 | 100 | 100 | |

Test Suites: 2 passed, 2 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 0.878s, estimated 1s
Ran all test suites.
```

Figura 19 – Jest

Após a instalação não é necessário usar mais nenhum método de configuração. É disponibilizada uma *API* própria desta *framework* em que através desta é feito o “*mock*” dos métodos em teste.

Este consegue correr vários testes de métodos e classes diferentes independentemente e em paralelo, mostrando depois o resultado destes como “*passed*” ou “*failed*” e onde estes falharam. O Jest através do comando `--coverage` consegue fazer também a cobertura geral de todas as classes e métodos que estão a ser testados, como na Figura 19.

3.3 Modeladores 3D

3.3.1 Blender 3D

Blender [24] é um *software open-source* de modelação e animação 3D (Figura 20). Este também possui funcionalidades para *rigging* (animação esquelética), escultura, simulação, renderização, captura de movimento, efeitos especiais, edição de vídeo, animação 2D e até alguma programação destes através de *scripts*.

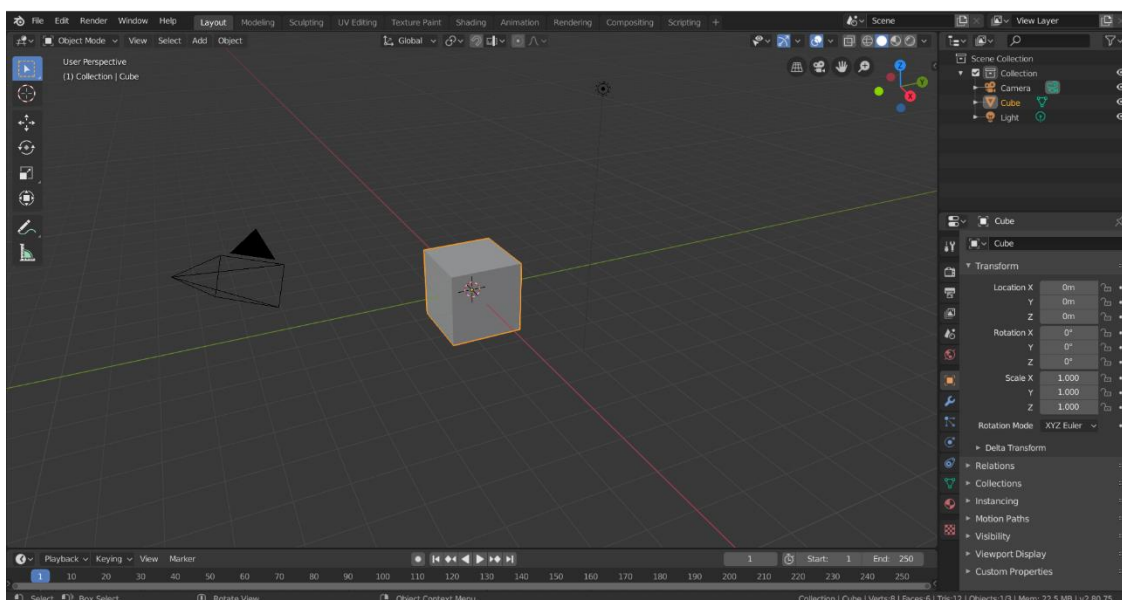


Figura 20 - Menu principal Blender

O Blender possui também bastante documentação, com guias e tutoriais para os mais diversos aspetos que podem ser aplicados neste *software* por parte dos utilizadores. Esta documentação inclui também alguns ficheiros de demonstração já prontos a serem usados no Blender. O Blender também possui uma vasta comunidade onde pode ser trocada informação entre diversos utilizadores, quer estes sejam estudantes, profissionais ou até entusiastas.

4 Design

Neste capítulo é apresentado o design de implementação da aplicação web desenvolvida no projeto.

4.1 Objetivos

Plataforma web de visualização de informação meteorológica em tempo real e 3D no mapa/localização escolhida pelo utilizador.

4.2 Requisitos

Os requisitos iniciais são descritos sob a forma de um diagrama de casos de uso, representado na Figura 21.

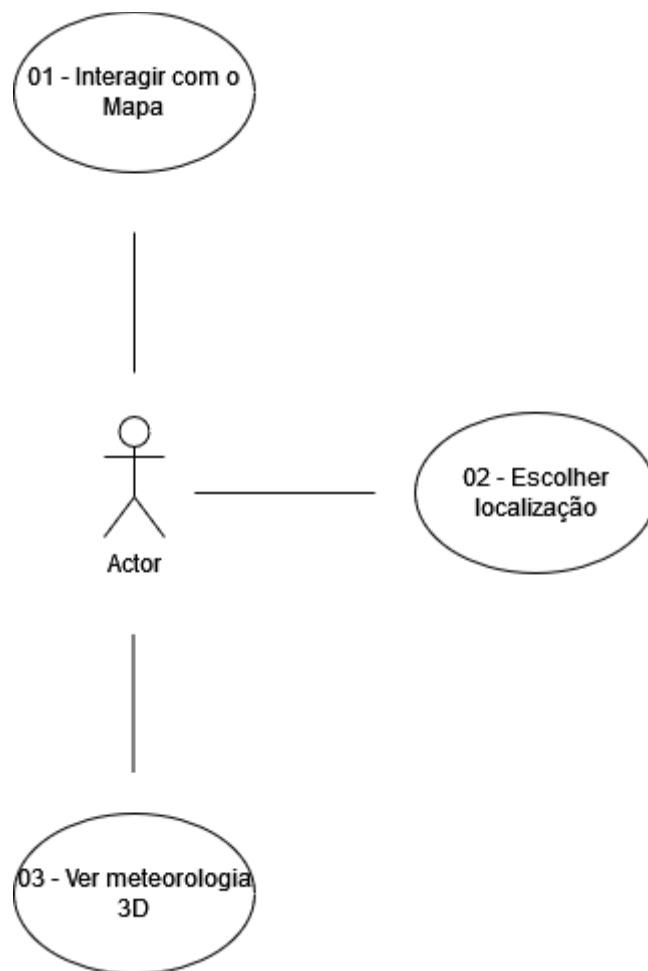


Figura 21 - Diagrama de Casos de Uso

4.2.1 UC01 – Interagir com o Mapa

O utilizador tem a possibilidade de visualizar um mapa na plataforma, mapa este onde irá ser mostrada informação meteorológica. Nas opções de visualização estão incluídas:

- **Mover o mapa;**
- **Fazer zoom;**
- Escolher o tipo de mapa (satélite, mapa normal, etc.);
- **Mudar o ângulo de visualização 3D;**

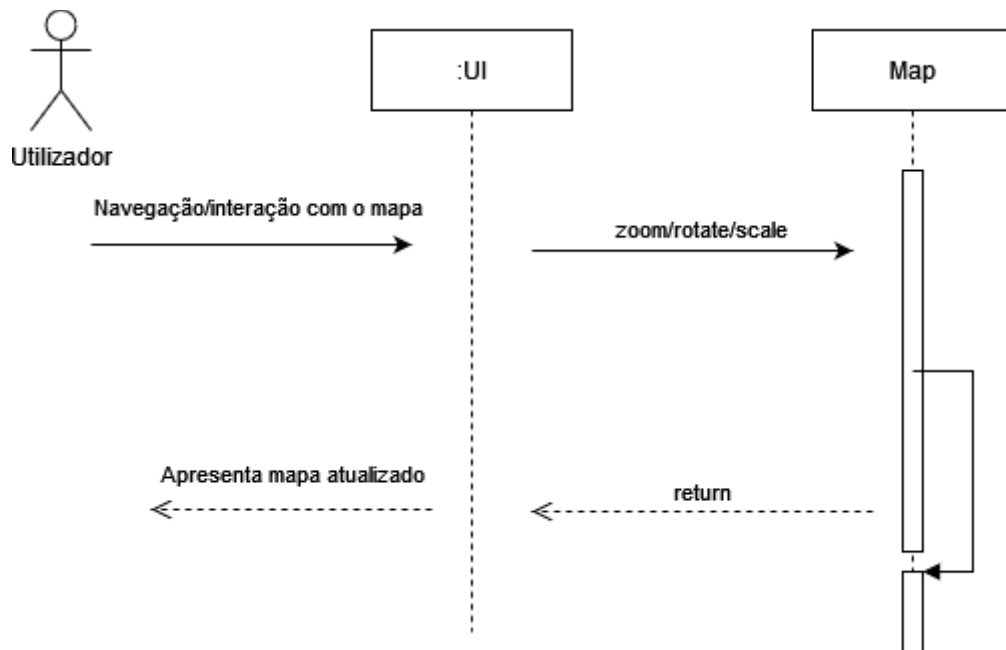


Figura 22 - Diagrama de sequência UC01

4.2.2 UC02 – Procurar Localização

O utilizador poderá escolher uma localização específica para obter informação detalhada sobre o estado do tempo naquele momento. Através de uma caixa de pesquisa, o utilizador pode indicar um local que quer ver no mapa.

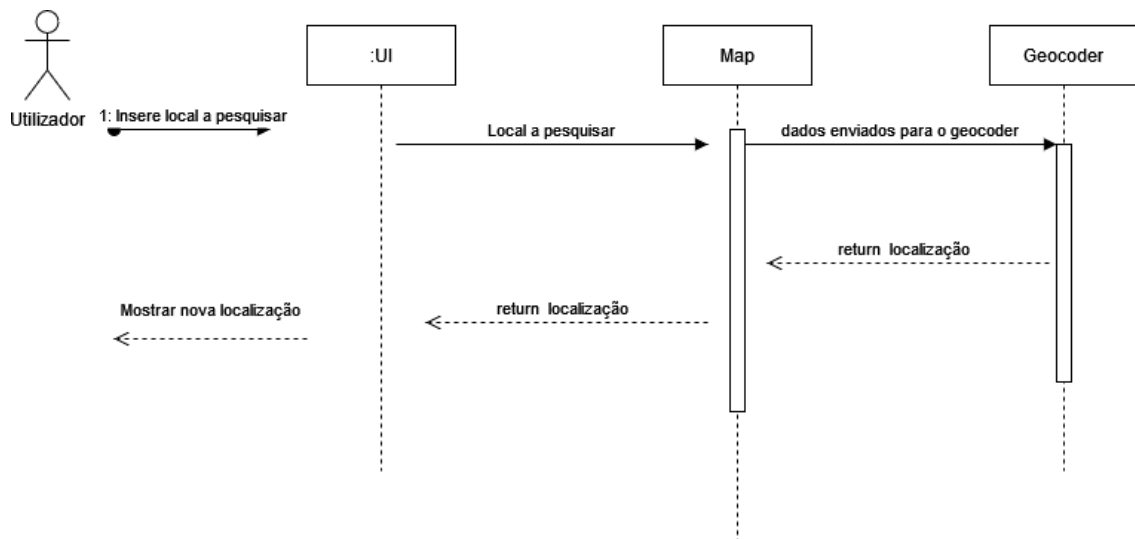


Figura 23 - Diagrama de sequência UC02

4.2.3 UC03 – Ver meteorologia 3D

É possível a visualização 3D de fenómenos meteorológicos. Por exemplo, se num dado local estiver a chover, é possível ver uma animação onde de facto está a ocorrer precipitação sobre um dado local. Também é permitida a escolha do tipo de informação a visualizar e a forma como a mesma é apresentada.

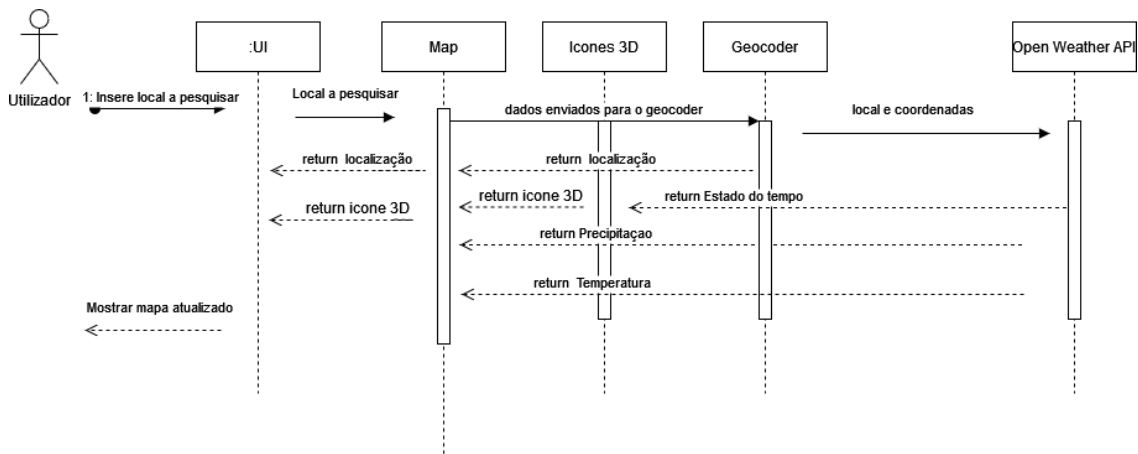


Figura 24 - Diagrama de sequência UC03

4.3 Limitações

- API com fornecimento de dados insuficientes/incapaz de fornecer os dados necessários para a aplicação a ser desenvolvida;
- poderá haver a incompatibilidade entre a plataforma web e os objetos 3D que vão ser importados na aplicação;
- as APIs gratuitas possuem limite de *requests* possíveis de fazer;
- limitações de performance 3D da plataforma usada;

4.4 Diagrama de Componentes

Após estudadas algumas das tecnologias que se enquadrariam neste projeto, foi concluído que só após a implementação do *webmap*, APIs de meteorologia e importação dos objetos 3D é que se poderia avançar com o resto do projeto. Estes são os “pilares” base do projeto na medida em que, sem estes, ficaria um produto inacabado que não apresentaria a qualidade desejada.

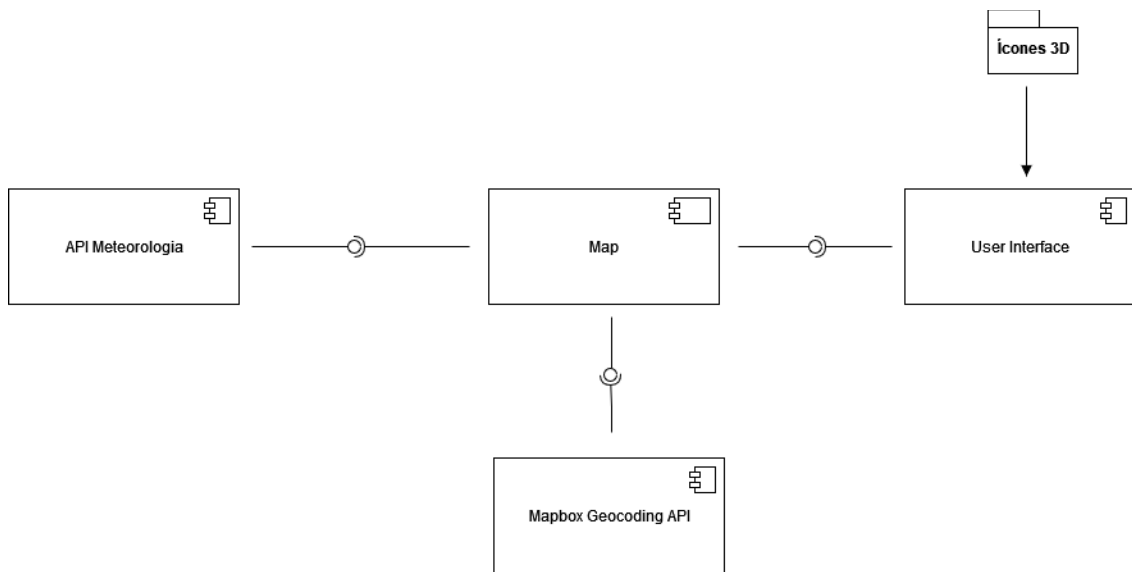


Figura 25 - Digrama de Componentes

Na figura acima é então representado o diagrama de componentes do projeto. Este é constituído por:

- API de Meteorologia – A API de meteorologia vai ser quem fornece os dados e informação sobre o estado do tempo;
- Map – Instância do próprio mapa;
- Mapbox Geocoding API – API de geocodificação do próprio mapbox;
- User Interface – Toda a interface com o utilizador, desde mapa, interação com este, “input” de texto para pesquisa, etc;
- Ícones 3D – Objetos 3D que vão ser importados e mostrados conforme a descrição do estado do tempo;

4.5 Soluções Alternativas

As soluções possíveis para este projeto baseiam-se num dos possíveis cenários:

- Gráficos com simbologia 3D dinâmica sobre o mapa;
- Gráficos de superfície 3D que se adaptam aos valores disponíveis sobre o mapa;

No primeiro, irá ser construído um mapa 2D em que através da importação dos ícones e animações 3D para a aplicação web, estes serão apresentados por cima de um local específico de acordo com o estado do tempo nesse local.

Já no segundo cenário, iria ser mostrado um mapa 3D com elevações de terreno onde consoante a variação de temperatura as cores do terreno iriam ter uma cor diferente adequada ao intervalo térmico nesse local.

5 Implementação

No capítulo 5 é feita a descrição de como a aplicação foi desenvolvida e implementada.

5.1 Estrutura

A estrutura da aplicação é bastante simples, contendo todos os ficheiros na mesma pasta. Assim, existe o ficheiro `weathermap3D.html`, que é o ficheiro principal. Este vai usar também os ficheiros `weathermap3D_style.css` e `weathermap3D_script.js`.

Dentro da mesma diretoria encontram-se também os ícones 3D e o ficheiro usado para os testes unitários.

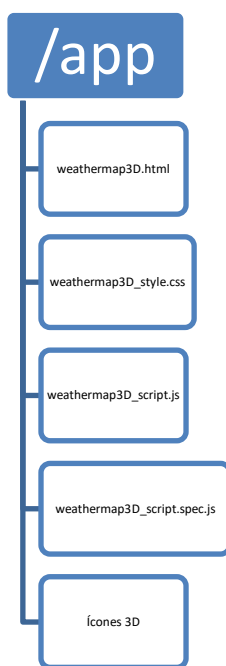


Figura 26 – Estrutura

Então, conforme a figura acima, os ficheiros que a aplicação usa são:

- `weathermap3D.html` – ficheiro base onde vai ser definida a estrutura do conteúdo da página web;
- `weathermap3D_style.css` – onde vai ser definida a aparência dessa mesma página;
- `weathermap3D_script.js` – define as funcionalidades e o comportamento da mesma;
- `weathermap3D_script.spec.js` – neste ficheiro estão escritos os testes unitários;

Vai ser então usado o *plugin* Mapbox GL JS Geocoder. Uma vez mais, é necessário adicionar ao ficheiro html o *link* para o JavaScript do *geocoder* como também o CSS. Para começar a usá-lo é adicionado o seguinte bloco de código:

```
var geocoder = new MapboxGeocoder({
  // Initialize the geocoder
  accessToken: mapboxgl.accessToken, // Set the access token
  mapboxgl: mapboxgl, // Set the mapbox-gl instance
  marker: false, // Do not use the default marker style
  placeholder: 'Pesquisar local', // Placeholder text for the search bar
});
```

Código 3 - Geocoder Mapbox

Para adicionar a caixa de pesquisa no mapa é usada a instrução `map.addControl(geocoder)`. O *MapboxGeocoder* vai fazer a conversão da *string* inserida na caixa de pesquisa para coordenadas geográficas.

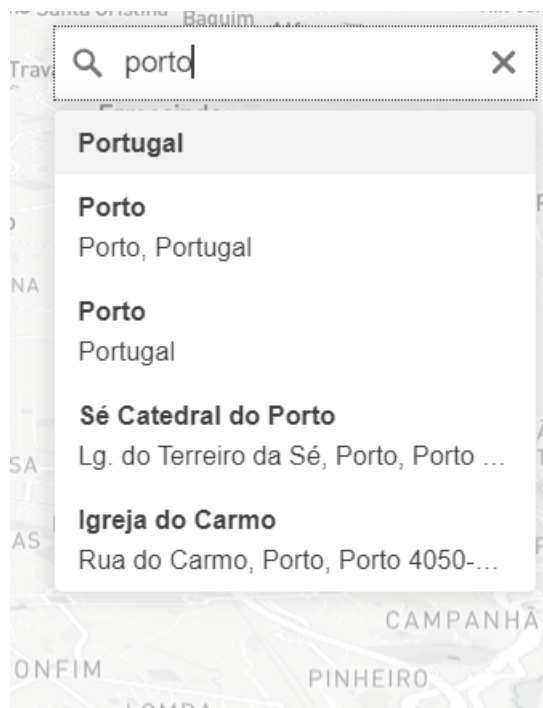


Figura 27 - Caixa de pesquisa

Ao detetar a escolha do local após um “*enter*” ou um “*click*” (Figura 27) a camera do mapa vai ser redirecionada para o local selecionado. A partir deste ponto é possível usar essa informação (retorno do local pesquisado) para o que é pretendido neste projeto: a recolha de dados

meteorológicos de um certo local a pesquisar e posterior informação visual.

5.3 API de meteorologia

A API escolhida para os serviços de meteorologia foi a OpenWeatherMap. Para usar esta API é também necessário criar uma conta e dessa forma também obter um *token* de acesso (Figura 28) que só através deste é possível começar a fazer chamadas á API e assim receber dados.

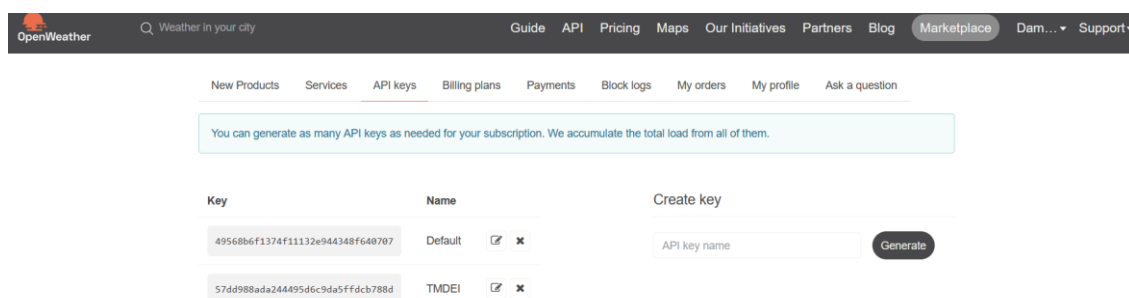


Figura 28 - Token de acesso OpenWeather

Após este passo é já possível começar a usar a API. Para este projeto é então necessário obter informações específicas para determinadas tarefas. Desta forma, como nem todas as chamadas á API devolvem tudo o que é necessário para apresentar as informações desejadas, é então preciso fazer três chamadas diferentes á API:

- Uma que no geral vai devolver o estado do tempo
- Outra para verificar a probabilidade de precipitação
- Uma outra para listagem de várias cidades e respetivo estado do tempo numa área de uma só vez

Para a primeira chamada, vai ser usado o local inserido pelo utilizador na barra de pesquisa de local para devolver então os dados meteorológicos desse mesmo local.

```

async function fetchmeteo(inputval,latitude,longitude)
{
  //var key = '57dd988ada244495d6c9da5ffdc788d';
  //'https://api.openweathermap.org/data/2.5/weather?id=2172797&appid=57dd988ada244495d6c9da5ffdc788d' -> funciona

  //Devolve precipitacao (pop - probability of precipitation)
  //https://api.openweathermap.org/data/2.5/onecall?lat=41.14&lon=-8.61&exclude=daily,minutely&appid=57dd988ada244495d6c9da5ffdc788d

  //Chamada API funcional com recolha da cidade/local inserido na barra de pesquisa
  //https://api.openweathermap.org/data/2.5/weather?q=' + inputval + '&appid=57dd988ada244495d6c9da5ffdc788d
  let f = await fetch('https://api.openweathermap.org/data/2.5/weather?q=' + inputval + '&appid=57dd988ada244495d6c9da5ffdc788d')
  .then(function (resp) { return resp.json() }) // Convert data to json
  .then(function (data) {
    //console.log(data);->funciona
    console.log('FETCHMETEO ');
    console.log("INFO -> "+data.weather[0].description);
    console.log(data);
  });
}

```

Código 4 - Chamada API do local pesquisado

Na figura acima é possível verificar que a variável `inputval` recebida como parâmetro vai ser usada para a construção dessa mesma chamada. A API vai então devolver o resultado em formato JSON, como mostra a Figura abaixo.

JSON	Dados em bruto	Cabeçalhos
Guardar	Copiar	Colapsar todos
Expandir tudo	Filtrar JSON	
▶ coord:	{...}	
▶ weather:	{...}	
base:	"stations"	
▶ main:	{...}	
visibility:	10000	
▶ wind:	{...}	
▶ clouds:	{...}	
dt:	1632699732	
▶ sys:	{...}	
timezone:	36000	
id:	2172797	
name:	"Cairns"	
cod:	200	

Figura 29 - Retorno API OpenWeather

Já para a verificação da precipitação, já foi necessário usar diretamente latitude e longitude na chamada á API, pois trata-se de uma chamada diferente que só desta forma é obtida a probabilidade de precipitação (Figura 30).

```

fetch('https://api.openweathermap.org/data/2.5/onecall?lat='+latitude+'&lon='+longitude+'&exclude=hourly,minutely&appid=57dd988ada244495d6c9da5ffdc788d')
.then(function (resp) { return resp.json() }) // Convert data to json
.then(function (data) {

```

Código 5 - Chamada API OpenWeather para precipitação

Como se verifica no código acima, aqui a chamada já é feita com o *onecall* e usa latitude e longitude, assim como são dados fornecidos á hora e ao minuto.

```
--  
▼ weather:  
  ▼ 0:  
    id:          803  
    main:        "Clouds"  
    description: "broken clouds"  
    icon:        "04n"  
    pop:         0.02
```

Figura 30 - Probabilidade de precipitação (pop)

Para o último tipo de chamada, é usada uma *bounding box* (caixa delimitadora) com as coordenadas limite norte, sul, este e oeste. Esta vai devolver as principais dez cidades dentro desta área.

A chamada é feita da seguinte maneira:
<https://api.openweathermap.org/data/2.5/box/city?bbox=-8.940811,41.027830,-8.287468,41.270066,12&appid=57dd988ada244495d6c9da5ffdcdb788d>

5.4 Ícones 3D

Para a questão dos ícones 3D foi desenvolvido um ícone diferente para cada um dos seguintes estados temporais:

- *clear sky*;
- *few clouds*;
- *scattered clouds*;
- *broken clouds*;
- *shower rain*;
- *rain*;
- *thunderstorm*;
- *snow*;
- *mist*;

A partir daqui foi modelado cada um destes ícones com recurso ao Blender 3D, representado na Figura abaixo.

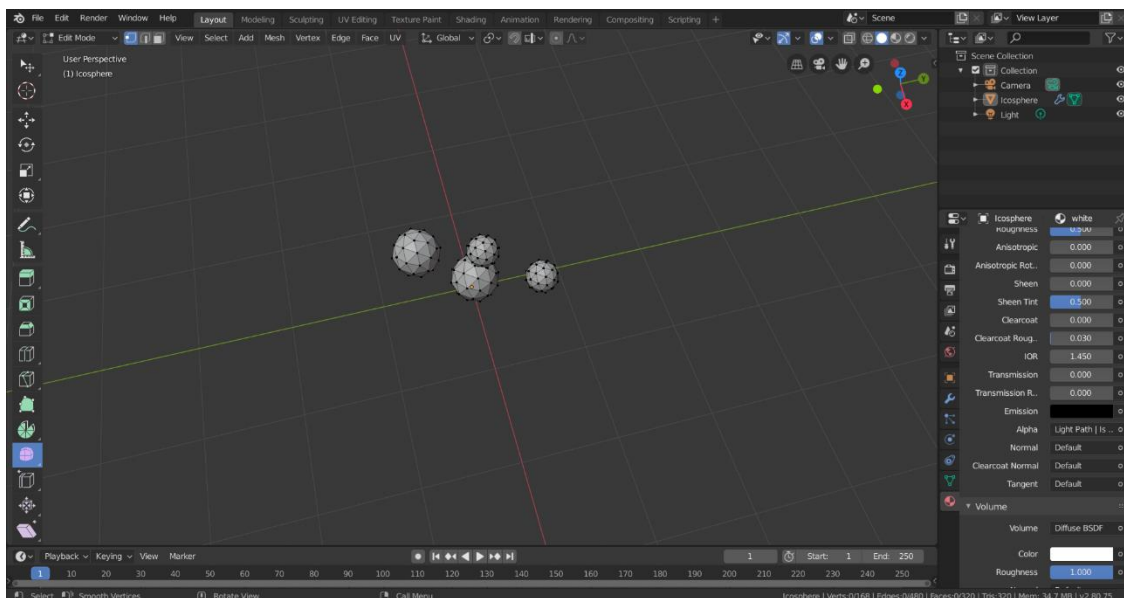

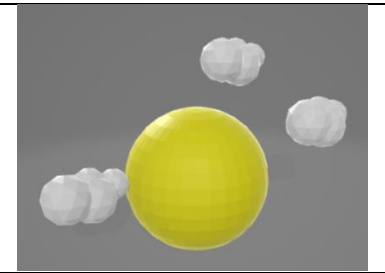
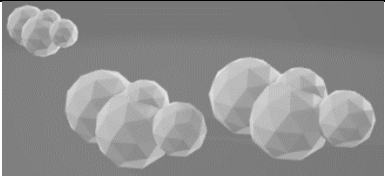
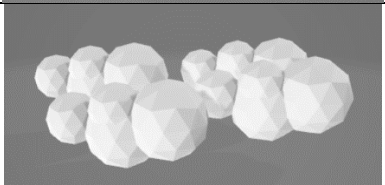






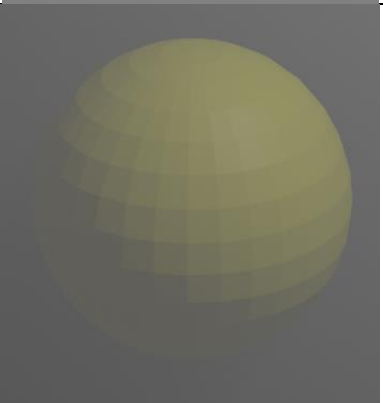
Figura 31 - Blender3D: Modelação de "broken clouds"

Foi então modelado um ícone para cada estado do tempo, como é possível ver na tabela abaixo. Por este tratar-se de um projeto-piloto, não foram criados ícones para versões noturnas dos objetos 3D em que aparece sol, em que neste caso apareceria uma lua.

Tabela 3 - Ícones 3D para cada estado do tempo

Descrição	Ícone 3D
clear sky	

few clouds	
scattered clouds	
broken clouds	
shower rain	
rain	

thunderstorm	
snow	
mist	

Como os ícones 3D vão ser desenhados com recurso ao *threeJS* e este só suporta formatos *glTF* (*Graphics Language Transmission Format*) é necessário então exportar os ficheiros em formato *blend* para *glTF* como na Figura 32.

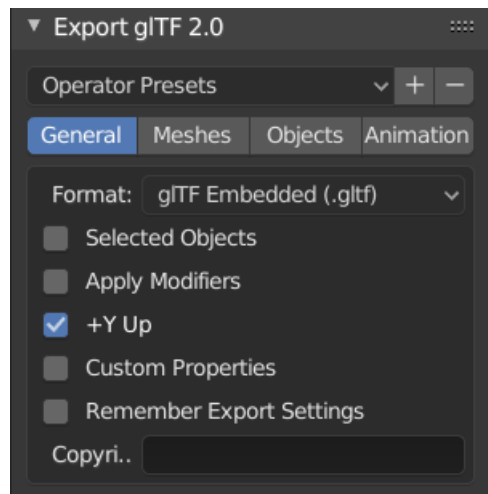


Figura 32 - Exportação para formato *glTF* no *Blender3D*

Tendo já todos os ícones modelados e no formato correto, é então necessário carregá-los na aplicação. O primeiro passo foi criar um método em que conforme cada estado do tempo, iria mostrar respectivo ícone:

```

async function meteo_conditions(d)
{
  console.log('HERE -> meteo_conditions()');
  console.log ('weather description - meteo_conditions -> '+d.weather[0].description);

  var wea,lon,lat = "";
  wea = d.weather[0].description;
  lon = d.coord.lon;
  lat = d.coord.lat;
  console.log ('wea ==>'+wea);

  if(wea == "clear sky")
  {
    console.log('HERE -> meteo_conditions() ->> clear sky');
    ico='./cs.glTF';
  }
  else if(wea == "overcast clouds")
  {
    console.log('HERE -> meteo_conditions() ->> overcast clouds');
    ico='./overcast_clouds.glTF';
  }
  else if(wea == "few clouds")
  {
    console.log('HERE -> meteo_conditions() ->> few clouds');
    ico='./few_clouds.glTF';
  }
}

```

Código 6 - Método `meteo_conditions()`

Este método vai receber a descrição do estado do tempo, identificar, comparar e então devolver o ícone correto.

5.5 Web Server



Figura 33 - Pasta raíz Web Server for Chrome

Para conseguir com que os objetos 3D fossem exportados para o mapa foi necessário recorrer a um servidor web. Visto que este projeto numa fase inicial, não precisa de estar alojado num

servidor web externo, a solução encontrada foi usar uma das várias extensões do Google Chrome, chamada Web Server for Chrome.

Um dos principais motivos para a necessidade de um servidor web foi relacionado com a política *Cross-Origin Resource Sharing* (CORS), que ocorre sempre que são usados ficheiros externos, neste caso esses ficheiros seriam os ícones 3D que iriam ser importados. Desta forma, a solução para ultrapassar esse problema foi encontrado com o Web Server for Chrome.

Ao lançar o Web Server, basta escolher a pasta onde está o projeto, como na Figura 17, e lançar o *url* predefinido. Como este é um servidor para uso interno/pessoal, não foi necessário abrir a ligação para o exterior.

Então após o arranque do servidor, basta escolher o ficheiro html (Figura 33) e app é lançada (Figura 34).

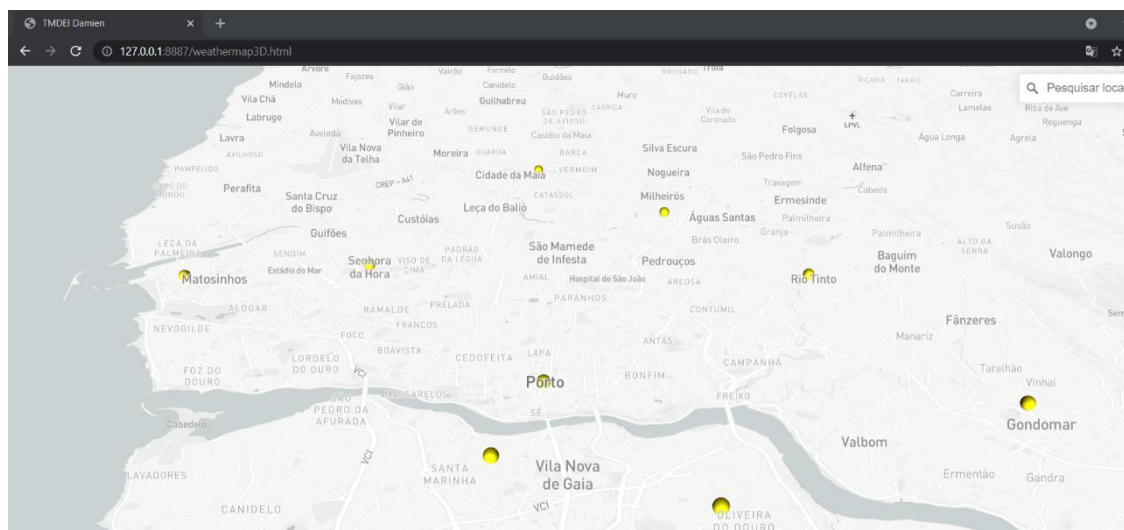


Figura 34 – Vista inicial mapa

5.6 Interface Gráfica

Toda a interface gráfica da aplicação é bastante simples e direta. Após abrir a página web o utilizador vai estar centrado na cidade do Porto e vai encontrar então o estado do empo na forma de ícones 3D em dez cidades nos arredores.

O único *input* direto de informação por parte do utilizador é a caixa de pesquisa onde este pode pesquisar a localidade pretendida. À parte da entrada de texto é possível ao utilizador, deslocar-se no mapa, através do arrasto do rato, fazer *zoom in* e *zoom out* assim como rodar o próprio mapa. Este também é capaz de manipular a inclinação do mapa, conseguindo, assim,

ter uma diferente perspetiva sobre o próprio mapa e os ícones 3D a serem visualizados naquele momento.

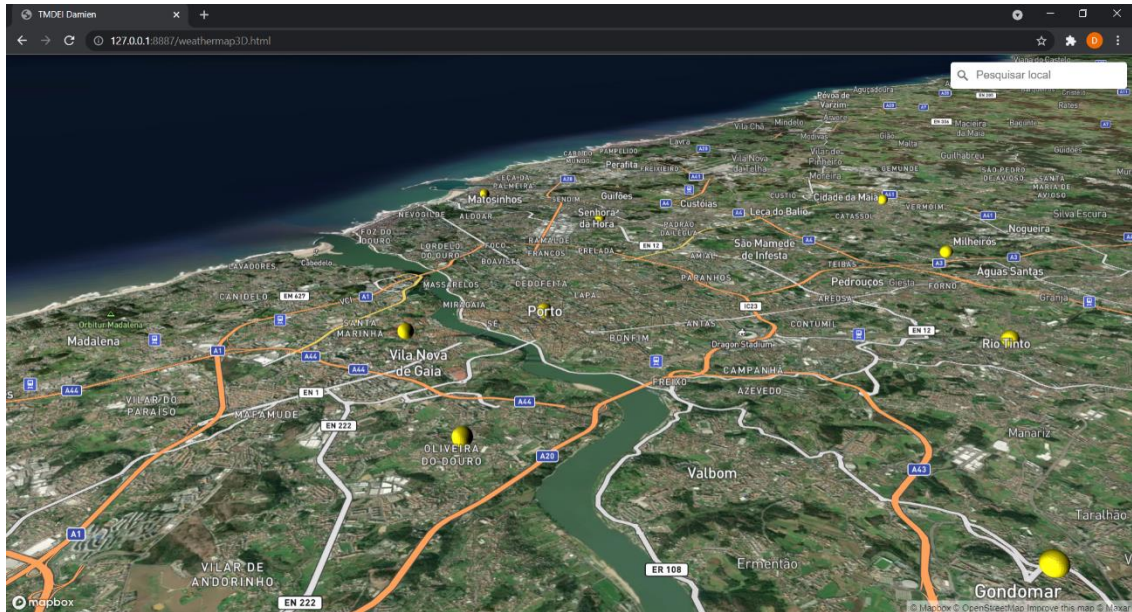


Figura 35 - Vista inicial satélite

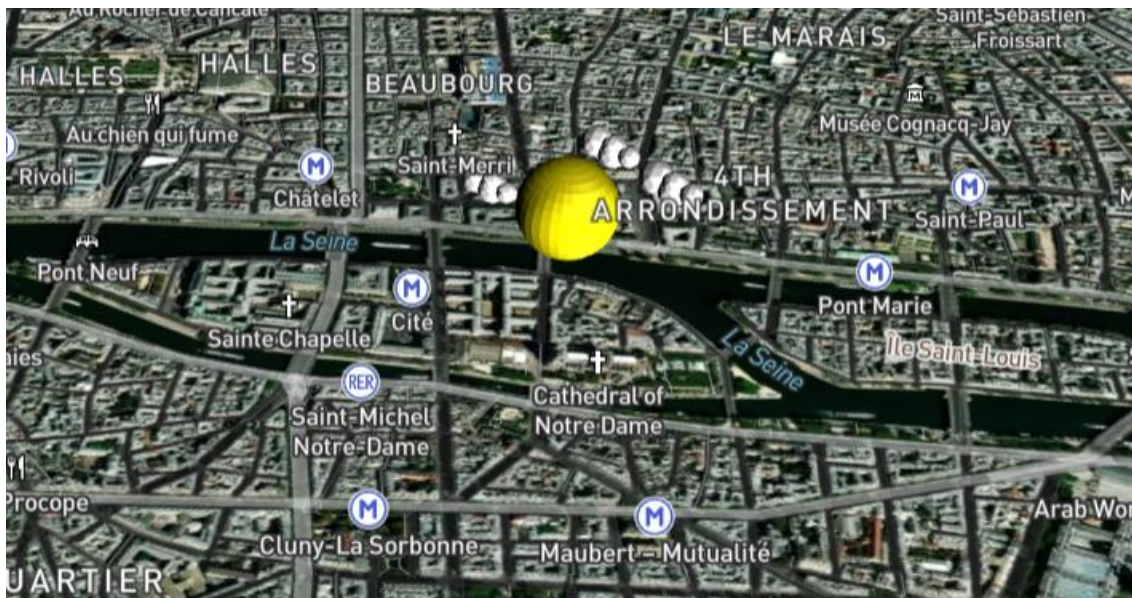


Figura 36 - Pesquisa pela cidade Paris

As figuras 35 e 36, mostram a vista inicial e o resultado de uma pesquisa quando com a vista satélite. Ícones diferentes vão aparecer conforme o estado do tempo naquele local específico. Enquanto ao pesquisar por um local isolado, o mapa vai recentrar o utilizador num dado local, como na Figura 36, em que o utilizador fez uma pesquisa por *'Paris'* e o mapa automaticamente redirecionou o utilizador para Paris.

Também após algum feedback por parte de utilizadores que testaram a aplicação, foi decidido colocar como vista predefinida a vista satélite, em vez de vista apenas de ruas como estava inicialmente representado na Figura 34.

6 Avaliação

No capítulo 6 é feita a descrição de como a aplicação foi testada e quais os testes feitos. Para os testes á aplicação foram realizados testes unitários, testes funcionais e testes de usabilidade.

6.1 Testes Unitários

Os testes unitários são realizados a partes do código-fonte. Para estes foi utilizada a *framework* de testes Jest.

Neste projeto foram feitos testes unitários com um maior foco nas verificações de chamadas á API de meteorologia.

```
async function fetchmeteo_rectangle_area() {
  const res = await fetch('https://api.openweathermap.org/data/2.5/box/city?bbox=-8.940811,41.027830,-8.287468,41.270066,12&appid=57dd988ada244495d6c9da5ffdc788d')
  const json = await res.json()

  return json
}

async function fetchmeteo() {
  const res = await fetch('https://api.openweathermap.org/data/2.5/weather?q=Lisboa&appid=57dd988ada244495d6c9da5ffdc788d')
  const json = await res.json()

  return json
}

async function fetchmeteo2() {
  const res = await fetch('https://api.openweathermap.org/data/2.5/onecall?lat=41.14&lon=-8.61&exclude=daily,minutely&appid=57dd988ada244495d6c9da5ffdc788d')
  const json = await res.json()

  return json
}
```

Código 7 - Funções testes unitários

Foram então criadas funções de teste, especificadamente para as chamadas á API, baseadas nas funções existentes no ficheiro `weathermap3D_script.js`. Uma vez mais, como são funções assíncronas, foi necessário recorrer á instrução *await*.

Após as funções criadas, é feito um “*mock*” ao *fetch*, e são finalmente efetuados os testes propriamente ditos. Estes começam com a instrução *describe*, onde o que esta instrução faz é criar um bloco que agrupa um conjunto de testes relacionados.

Então dentro deste bloco, são criados os testes. Cada um destes testes começa com o método *test* em que dentro destes é então verificado se o teste que está a ser feito, está a fazer o que é suposto, assim como também obtém o resultado esperado. Para então verificar estes mesmos critérios e condições é usado o *expect*. Através deste é possível associar um vasto número de “*matchers*” que faz com que seja possível associar ao resultado pretendido. Um bloco de testes pode ser visto na figura seguinte.

```

describe('fetchmeteo_rectangle_area', () => {
  test('works', async () => {
    const json = await fetchmeteo_rectangle_area()
    expect(Array.isArray(json)).toEqual(true)
    expect(json.length).not.toBeNull()
  })
})

describe('fetchmeteo', () => {
  test('works', async () => {
    const json = await fetchmeteo()
    expect(Array.isArray(json)).toEqual(true)
    expect(json.length).not.toBeNull()
  })
})

describe('fetchmeteo2', () => {
  test('works', async () => {
    const json = await fetchmeteo2()
    expect(Array.isArray(json)).toEqual(true)
    expect(json.length).not.toBeNull()
  })
})

```

Código 8 - Bloco de testes unitários

Após os testes criados, basta abrir a linha de comandos na localização onde este mesmo ficheiro de testes se encontra e executar o comando: `npm run test` (Figura 35)

```

PS C:\Users\asus\Desktop\ISEP\20_21\TMDEI\GIT\tmdei\app> npm run test
> app@1.0.0 test C:\Users\asus\Desktop\ISEP\20_21\TMDEI\GIT\tmdei\app
> jest
PASS ./weathermap3D_script.spec.js
  ✓ passes when value is NaN (1 ms)
  fetchmeteo_rectangle_area
    ✓ works (5 ms)
  fetchmeteo
    ✓ works
  fetchmeteo2
    ✓ works

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        0.408 s, estimated 1 s
Ran all test suites.

```

Figura 37 - Testes unitários efetuados com sucesso

6.2 Testes Funcionais

Os testes funcionais baseiam-se no próprio funcionamento da aplicação, isto é, se uma certa funcionalidade realmente está a funcionar.

Para os testes funcionais foi criada a seguinte tabela onde é verificado se as funcionalidades definidas pelo diagrama de casos de uso foram cumpridas.

Tabela 4 - Testes Funcionais

Teste Funcional	Resultado
TF01 – Mover Mapa	OK
TF02 – Fazer Zoom	OK
TF03 – Escolher tipo de mapa	Não implementado
TF04 – Mudar ângulo de visualização 3D	OK
TF05 – Procurar Localização	OK
TF06 – Ver meteorologia 3D	OK
TF07 – Arranque do mapa num local predefinido	OK

Como é possível verificar na tabela anterior, a maioria dos testes funcionais foram cumpridos com sucesso. Apenas a questão da escolha do tipo de mapa acabou por não ser implementado pois era algo secundário para o projeto.

6.3 Testes de Usabilidade

Os testes de usabilidade foram efetuados perante um pequeno grupo de pessoas para verificar o quão intuitivo é o uso da aplicação.

Para tal, foi criado um guião com os seguintes passos:

1. Ao abrir o mapa, vai encontrar o mapa centrado no Porto, com o estado do tempo assim como nas cidades circundantes.
2. Através dos controlos do rato, desloque-se até Lisboa.
3. Através da caixa de pesquisa, verifique o tempo em Beja.
4. Usando a tecla ctrl + rato rode o mapa assim como a inclinação do mesmo.
5. Verifique o estado do tempo numa localidade á sua escolha.

Foi também criado um formulário para que os inquiridos pudessem responder e dar o seu feedback sobre a aplicação.

O inquérito foi respondido por pessoas de ambos os sexos, com idades compreendidas entre os 26 e os 63 anos. Entre estes utilizadores, o nível de experiência de interação com o computador vai desde utilizadores avançados, como engenheiros informáticos, até pessoas mais inexperientes que possuem poucos conhecimentos de como usar um computador. Foram obtidas 8 respostas.

O feedback e respostas obtidas foram os seguintes:

O mapa é de fácil navegação?
8 respostas

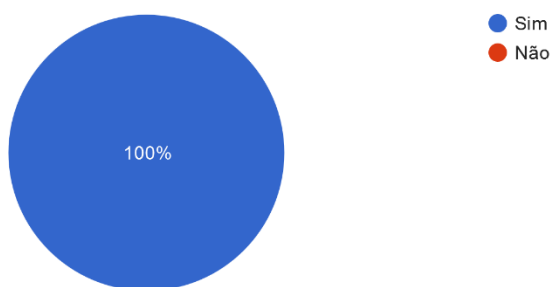


Figura 38 - Formulário pergunta nr.1

À pergunta sobre quão fácil é a navegação no mapa, Figura 38, este teve um resultado de 100%.

Conseguiu pesquisar o local pretendido?
8 respostas



Figura 39 - Formulário pergunta nr.2

Quanto á pesquisa de locais, Figura 39, todos os inquiridos não tiveram dificuldades em pesquisar o local pretendido.

Conseguiu fazer zoom in/out (ampliar/diminuir) o mapa)
8 respostas

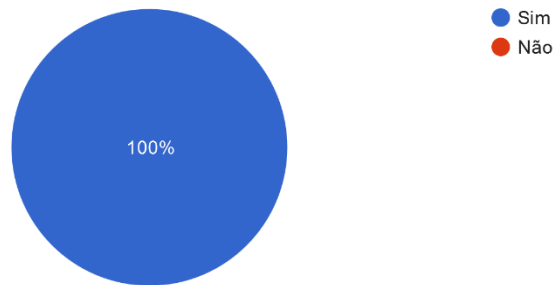


Figura 40 - Formulário pergunta nr.3

Conseguiu rodar o mapa?
8 respostas

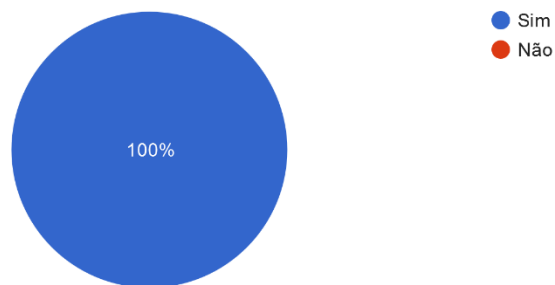


Figura 41 - Formulário pergunta nr.4

Também quando á apresentação do mapa, rodar, ampliar e diminuir (Figuras 40 e 41) todos os utilizadores que utilizaram a aplicação não tiveram problemas em fazê-lo.

Os ícones de estado do tempo são compreensíveis?

8 respostas

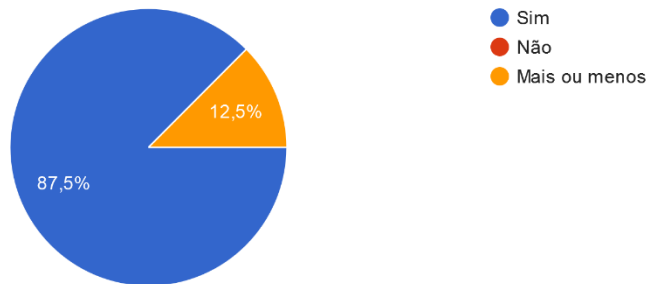


Figura 42 - Formulário pergunta nr.5

Quanto á compreensão dos ícones apresentados, na Figura 42, 12,5% dos inquiridos não interpretaram logo de forma correta o significado dos ícones.

O que gostou mais do sistema?

6 respostas

- Fácil procura de informação
- Interação 3D
- O mapa é muito compreensível.
- Facilidade de navegação
- Da apresentação.
- rapidez

Figura 43 - Formulário pergunta nr.6

Relativamente ao que os utilizadores mais gostaram no sistema, foi na sua generalidade a própria interface do mapa, desde navegação até á própria pesquisa de locais, como é possível verificar na Figura 43.

Tem alguma sugestão para melhorar o sistema

3 respostas

Modo offline

Vista satélite

Adicionar opção para escolha do tipo de mapa(2D, 3D, satélite)

Figura 44 - Formulário pergunta nr.7

Quanto a sugestões a melhorar, na Figura 44, os utilizadores responderam a possibilidade de mudar o tipo de vista do mapa como a principal funcionalidade em falta.

7 Conclusões

Neste capítulo são apresentadas as conclusões finais sobre o projeto realizado. São descritos os objetivos realizados e uma descrição sobre o seu estado. São também mostradas as limitações do trabalho efetuado, e por fim, é apresentada uma apreciação pessoal sobre o trabalho desenvolvido.

7.1 Objetivos realizados

Quanto aos objetivos realizados no projeto, todos os objetivos principais foram cumpridos, á exceção da escolha do tipo de mapa e das colunas em forma de gráfico sobre o mapa, onde teria a indicação da probabilidade de precipitação naquela zona específica. Funcionalidades como a navegação do mapa, pesquisa de localização ou a própria visualização 3D do estado do tempo foram implementados com sucesso.

7.2 Limitações e trabalho futuro

Por este tratar-se de um projeto-piloto, houve várias funcionalidades que seriam interessantes implementar e foram descartadas. Uma delas foi a questão da implementação de colunas em forma de gráfico sobre o mapa para verificar a probabilidade de precipitação de um dado local. Aqui também poderia ser utilizado texto 3D com a temperatura atual na zona escolhida.

Isto poderia ser enquadrado no mapa de duas formas:

- Ao clicar no ícone do estado do tempo surgiria por cima do ícone um *popup* com a coluna de probabilidade de precipitação e o texto 3D da temperatura atual;
- Ao inicializar o mapa existiriam botões onde seria possível escolher o tipo de informação meteorológica a ser visualizada pelo utilizador;

Também seria interessante melhorar e animar os ícones 3D. Desde já, criar uma versão noturna para os ícones que apresentem um sol e depois disso criar uma versão mais realista dos mesmos e não uma versão *lowpoly*⁶.

O passo seguinte seria a própria animação dos ícones até no momento em que o utilizador estivesse a usar a aplicação e o estado do tempo passasse de ensolarado para nublado, este via a transição dos ícones. Uma funcionalidade interessante a implementar também seria a hipótese de se poder escolher o tipo de mapas (vista satélite, modo noturno, ruas, etc.), através de botões inseridos na interface com o utilizador.

⁶ Malha poligonal em computação gráfica 3D com baixa quantidade de polígonos

7.3 Apreciação final

Este foi um projeto interessante, em que apesar de ter havido algumas limitações a nível de tecnologia e implementação, seria algo que com mais algum trabalho e melhorias, poderia vir a ter um bom potencial para vir a ser implementado em várias situações.

Referências

- [1] RapidAPI Staff, "Top 8 Best Free Weather APIs to Access Global Weather Data (Updated for 2021)," [Online]. Available: <https://rapidapi.com/blog/access-global-weather-data-with-these-weather-apis/>. [Acedido em 2 Fevereiro 2021].
- [2] Cara Hogan, "The Top 9 Weather APIs for 2021," [Online]. Available: <https://www.climacell.co/blog/top-8-weather-apis-for-2020/>. [Acedido em 2 Fevereiro 2021].
- [3] Shyam Purkayastha, "Top 10 Best Weather APIs," [Online]. Available: <https://blog.api.rakuten.net/top-weather-apis/>. [Acedido em 2 Fevereiro 2021].
- [4] D. S. API, "DarkSky," [Online]. Available: <https://darksky.net/dev>.
- [5] "Weather API," [Online]. Available: <https://openweathermap.org/api>. [Acedido em 7 Fevereiro 2021].
- [6] "AccuWeather," [Online]. Available: <https://developer.accuweather.com/>. [Acedido em 11 Fevereiro 2021].
- [7] "ClimaCell," [Online]. Available: <https://www.tomorrow.io/>. [Acedido em 8 Março 2021].
- [8] "weatherstack," [Online]. Available: <https://weatherstack.com/>. [Acedido em 9 Março 2021].

- [9] "Plataforma Google Maps," [Online]. Available: <https://developers.google.com/maps>. [Acedido em 25 Junho 2021].
- [10] "Google Maps API," [Online]. Available: <https://cloud.google.com/maps-platform/>. [Acedido em 25 Junho 2021].
- [11] "Leaflet," [Online]. Available: <https://leafletjs.com/>. [Acedido em 25 Junho 2021].
- [12] "Mapbox GL JS," [Online]. Available: <https://docs.mapbox.com/mapbox-gl-js/guides/>. [Acedido em 6 Junho 2021].
- [13] John Corpuz, "Best weather apps for 2021," [Online]. Available: <https://www.tomsguide.com/round-up/best-weather-apps>. [Acedido em 15 Fevereiro 2021].
- [14] Max Eddy & Jordan Minor, "The Best Weather Apps for 2021," [Online]. Available: <https://www.pcmag.com/picks/best-weather-apps>. [Acedido em 15 Fevereiro 2021].
- [15] Apple Inc., "Dark Sky," [Online]. Available: <https://darksky.net/app>. [Acedido em 15 Fevereiro 2021].
- [16] "Windy," [Online]. Available: <https://www.windy.com/>. [Acedido em 2 Março 2021].
- [17] "Foreca," [Online]. Available: <https://www.foreca.com/pt/>. [Acedido em 2 Março 2021].
- [18] "The Weather Channel," [Online]. Available: <https://weather.com/>. [Acedido em 7 Março 2021].
- [19] "Web Server for Chrome," [Online]. Available: <https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhemlocgigb>. [Acedido em 30 Agosto 2021].
- [20] "three.js," [Online]. Available: <https://threejs.org/>. [Acedido em 10 Agosto 2021].
- [21] "node.js," [Online]. Available: <https://nodejs.org/en/>. [Acedido em 1 Agosto 2021].
- [22] "npm," [Online]. Available: <https://www.npmjs.com/>. [Acedido em 2 Setembro 2021].
- [23] "Jest," [Online]. Available: <https://jestjs.io/>. [Acedido em 2 Setembro 2021].
- [24] "blender," [Online]. Available: <https://www.blender.org/>. [Acedido em 5 Julho 2021].
- [25] I. AccuWeather, "AccuWeather," 2021. [Online]. Available: <https://developer.accuweather.com/>.

[26] Max Eddy & Jordan Minor, "The Best Weather Apps for 2021," [Online]. Available: <https://www.pcmag.com/picks/best-weather-apps>.

Anexo - Questionário

Visualização 3D interactiva de informação meteorológica georreferenciada

Inquérito de usabilidade

O mapa é de fácil navegação? *

Sim

Não

Conseguiu pesquisar o local pretendido? *

Sim

Não

Mais ou menos

Conseguir fazer zoom in/out (ampliar/diminuir) o mapa) *

Sim

Não

Conseguir rodar o mapa? *

Sim

Não

Os ícones de estado do tempo são compreensíveis? *

Sim

Não

Mais ou menos

O que gostou mais do sistema?

Texto de resposta longa

Tem alguma sugestão para melhorar o sistema

Texto de resposta longa
