



UNIVERSIDADE D
COIMBRA

Manuel Tolda Martins Carrilho Fideles

PLATAFORMA DE VISUALIZAÇÃO MODULAR
DESENVOLVIMENTO DE UMA FERRAMENTA ANALÍTICA
PARA ESTUDOS DAS CONDIÇÕES CLIMATÉRICOS

Dissertação no âmbito do Mestrado em Engenharia e Ciência dos Dados, orientada pelo Professor Evgheni Polisciuc e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro 2023



DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Manuel Tolda Martins Carrilho Fideles

PLATAFORMA DE VISUALIZAÇÃO MODULAR
DESENVOLVIMENTO DE UMA FERRAMENTA
ANALÍTICA PARA ESTUDOS DAS CONDIÇÕES
CLIMATÉRICOS

Dissertação no âmbito do Mestrado em Engenharia e Ciência dos Dados,
orientada pelo Professor Evgheni Polisciuc e apresentada ao Departamento de
Engenharia Informática da Faculdade de Ciências e Tecnologia da
Universidade de Coimbra.

Setembro 2023



DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Manuel Tolda Martins Carrilho Fideles

MODULAR VISUALIZATION PLATFORM
DEVELOPMENT OF AN ANALYTICAL TOOL FOR
WEATHER STUDIES

Dissertation within the scope of the Master's Degree in Data Science and Engineering, supervised by Professor Evgheni Polisciuc and presented to the Department of Computer Engineering of the Faculty of Science and Technology of the University of Coimbra.

September 2023

Agradecimentos

Aos meus pais, pela educação e apoio incondicional.

Às minhas irmãs, por serem sempre as minhas melhores amigas.

Aos meus amigos, por aturarem os meus caprichos e por me fazerem rir.

À minha avó, pelos lanches e carinho.

E ao meu avô, que olha por mim lá de cima.

Resumo

A modularidade é uma tendência cada vez mais relevante no desenvolvimento para a *web*, com a solidificação no mercado de tecnologias como o ReactJS, o VueJS, e AngularJS a liderar a compartimentalização, separação de preocupações e, acima de tudo, a facilidade de manutenção do código subjacente. A área de visualização não é exceção. Começa a ser cada vez mais notório o uso dessas tecnologias, abrindo novas possibilidades de paradigmas de visualização, particularmente em aplicações avançadas. A visualização analítica, que é útil para dados complexos e heterogêneos, pode usufruir dos benefícios da modularidade. Todavia, a integração destas técnicas numa única ferramenta é difícil devido às especificidades conceituais e de implementação de cada visualização. Este facto conduz à necessidade de uma plataforma que agregue múltiplos tipos de visualizações e dados, de forma modular. Este estudo visa demonstrar a viabilidade de uma solução modular no contexto da visualização analítica, aplicada à dinâmica atmosférica de uma determinada localização urbana. Assim, foi desenvolvida uma plataforma de visualização de informação geográfica, integrando ferramentas de visualização analíticas, que solidifica os conhecimentos prévios acerca da atmosfera inferior numa dada localidade, e que potencia a extração de novas conclusões acerca dos fenómenos climáticos ali existentes. Para além disso, os testes de utilizador levados a cabo para validar a plataforma desenvolvida demonstram a eficácia da aplicação do paradigma de modularidade em aplicações de visualização avançada.

Palavras-Chave

Visualização de Dados, Visualização Geográfica, Visualização Analítica, Análise Espaço-temporal, Visualização para Web, D3.js, React.js.

Abstract

Modularity is an increasingly relevant trend in development for the *web*, with the solidification in the market of technologies such as ReactJS, VueJS, and AngularJS leading the way in compartmentalization, separation of concerns, and, above all, ease of maintenance of the underlying code. The data visualization field is no exception. The use of these technologies is becoming increasingly noticeable, opening up new possibilities for visualization paradigms, particularly in advanced applications. Analytical visualization, which is helpful for complex and heterogeneous data, can enjoy the benefits of modularity. However, integrating these techniques into a single tool is difficult due to each visualization's conceptual and implementation specificities. This leads to the need for a modular platform aggregating multiple types of visualizations and data. This study aims to demonstrate the viability of a modular solution in the context of analytical visualization, applied to the atmospheric dynamics of a given urban location. Thus, a geographic information visualization platform was developed, integrating analytical visualization tools, which solidifies previous knowledge about the lower atmosphere in a given area, and enhances the extraction of new conclusions about the climatic phenomena that happen there. In addition, user tests were carried out to validate the platform developed to demonstrate the effectiveness of applying the modularity paradigm to advanced visualization applications.

Keywords

Data Visualization, Geographical Visualization, Visual Analytics, Spatio-temporal Analysis, Visualization for Web, D3.js, React.js.

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	2
1.3	Âmbito e Objetivos	3
1.4	Estrutura do Documento	4
2	Estado da Arte	5
2.1	Visualização Geográfica	5
2.1.1	Símbolos Cartográficos	6
2.1.2	Mapas de Pontos	9
2.1.3	Mapas Coropléticos	12
2.1.4	Mapas de Superfície	12
2.1.5	Outros Mapas	15
2.2	Visualização Analítica	19
2.2.1	Dashboards	21
2.3	Tarefas de Utilizador	21
2.4	Técnicas de Interação	24
2.4.1	Overview plus Detail	24
2.4.2	Linking and Brushing	25
2.4.3	Focus plus Context	25
2.4.4	Panning and Zooming	26
2.5	Tecnologias	27
2.5.1	Pré-processamento de dados	27
2.5.2	Backend-as-a-Service	29
2.5.3	Frameworks de <i>UI</i>	31
2.5.4	Visualização	36
3	Trabalho Relacionado	41
3.1	COVID19INDIA	41
3.1.1	Objetivos	41
3.1.2	A Solução	41
3.1.3	Forças e Fraquezas	43
3.2	Banda Ultra Larga Italia	46
3.2.1	Objetivos	46
3.2.2	A Solução	46
3.2.3	Forças e Fraquezas	48
3.3	UCLA Energy Atlas	48
3.3.1	Objetivos	50

3.3.2	A Solução	50
3.3.3	Forças e Fraquezas	51
3.4	Síntese	51
4	Objetivos e Metodologias	55
4.0.1	Objetivos	55
4.0.2	Metodologia	56
5	Trabalho Prático	57
5.1	Planeamento e Gestão de Projeto	57
5.1.1	Requisitos e análise de risco	57
5.1.2	Gestão do progresso	64
5.2	Dados	66
5.2.1	Caracterização	66
5.2.2	Modelação	68
5.2.3	Pré-processamento e armazenamento	69
5.2.4	Dificuldades e desafios	69
5.2.5	Tecnologias utilizadas	71
5.3	Plataforma	71
5.3.1	Arquitetura	72
5.3.2	Páginas	73
5.3.3	Dificuldades e desafios	87
5.3.4	Tecnologias utilizadas	88
5.4	Validação	89
5.4.1	Metodologia	89
5.4.2	Resultados	90
5.4.3	Análise e Reflexão	93
6	Conclusão	95
6.1	Trabalho Realizado	96
6.2	Trabalho Futuro	96
	Referências	99
	Apêndice A Requisitos Funcionais	105
	Apêndice B Testes de Usabilidade	107
B.1	Documento de Consentimento Informado	107
B.2	Questionário Pré-Teste	107
B.3	Tarefas	108
B.4	Questionário Pós-Teste	110

Lista de Figuras

2.1	Captura de ecrã de uma porção da interface do GLEAMViz, uma plataforma de controlo e análise da propagação de doenças. As ocorrências de uma determinada doença são representadas por pontos vermelhos sobre o mapa [GLEAM, 2022].	6
2.2	Mapa de símbolos em React-Mapbox. Os pontos são agregados em <i>clusters</i> conforme o nível de zoom. Para além disso, as cores dos <i>clusters</i> são modificadas conforme o número de elementos que contêm - a cor rosa corresponde a um maior número de elementos, e a cor azul corresponde a um menor número de elementos [vis.gl, 2017].	7
2.3	Linhas isotérmicas de Humboldt. Neste exemplo, as linhas são utilizadas para ligar os locais com a mesma temperatura média anual [Robinson and Wallis, 1967].	8
2.4	Estudo de Loomis sobre duas tempestades que ocorreram em Fevereiro de 1842. A área é utilizada para comunicar diferentes estados metereológicos [Loomis, 1845].	9
2.5	Captura de ecrã da interface do LAEI-NO2. Neste exemplo, o volume é utilizado para representar a concentração de dióxido de nitrogénio [Parallel, 2020]	9
2.6	Mapa de mortes causadas pela bactéria da cólera em Londres, por Snow. Este estudo mapeia as localização geográficas - representadas por pontos - das incidências da incidência de cólera em Londres [Tennekes, 2018].	10
2.7	Mapa das ocorrência da febre Q na Holanda. À semelhança do mapa do Dr. Snow, as localizações geográficas de ocorrências da doença são representadas por pontos vermelhos sobre um mapa da Holanda [Soetens et al., 2017].	11
2.8	Mapeamento das infeções por <i>Schistosoma mansoni</i> no Brasil [Barreto, 1993].	11
2.9	Mapa do analfabetismo em França. Neste mapa, as várias tonalidades de cinza representam o grau de analfabetismo (uma variável categórica) nas unidades administrativas do país - tons mais escuros representam maior analfabetismo [Dupin, 1826].	13
2.10	Mapa da prostituição em França [Parent-Duchatelet, 1836].	13
2.11	Mapa do progresso da implementação de redes wireless e fibra em Itália. As tonalidades coloridas no mapa indicam a percentagem de locais de instalação de fibra que terminaram o seu trabalho [BUL, 2020].	14

2.12	Mapa de linhas isotérmicas de Woodbridge [Woodbridge, 1823]. . .	15
2.13	Precipitação na região oriental de Espanha, a Junho de 2000 [Centre, 2022].	16
2.14	Alojamento turístico nas ilhas Baleares em 1985, 1996, e 2010 [Pons et al., 2014].	16
2.15	Exportações de vinho francesas em 1864, por Minard [Minard, 1869].	17
2.16	Produção de carne em França [Friendly, 2002].	18
2.17	Cartograma das eleições presidenciais de 2020. Neste exemplo, a área de cada estado é distorcida de acordo com o seu número de habitantes [WorldMapper, 2020].	19
2.18	Disciplinas que compõem a visualização analítica [Keim and Zhang, 2011].	20
2.19	O ciclo de aquisição e geração de conhecimento [Keim et al., 2008].	20
2.20	Amazon QuickSight [Bach et al., 2022].	22
2.21	COVID19INDIA [Bach et al., 2022].	22
2.22	<i>Dashboard do Instagram Analytics</i> [Bach et al., 2022].	23
2.23	Captura de ecrã do editor de texto VSCode. Neste exemplo, temos uma visão geral do código todo sob a forma de um mini-mapa (do lado direito) e de uma secção particular do lado esquerdo.	25
2.24	Barra de tarefas do MacOS. Neste exemplo, é possível dar maior destaque a um ícone específico, mantendo em vista os restantes ícones que compõem a barra de tarefas [Cockburn et al., 2009]. . . .	26
2.25	Técnica de <i>fish-eye lens</i> [KloECKl et al., 2012].	26
2.26	Página principal da interface do Google Maps. Esta plataforma permite ao utilizador arrastar com o rato o seu campo de visão para outras localizações geográficas, focando em áreas diferentes. Para além disso, é ainda possível fazer zoom-in ou zoom-out, aumentando ou reduzindo a abrangência do seu campo de visão. . . .	27
2.27	Exemplo do mesmo elemento React criado com recurso a JavaScript puro (em cima), e com JSX (em baixo)	32
2.28	Atualização da DOM através da DOM virtual. O elemento <i>img</i> mudou, e portanto é atualizada na DOM real. Apenas este componente é atualizado, com o resto dos componentes da página a manterem-se inalterados [Fedosejev, 2015].	33
2.29	Exemplo da decomposição da interface de utilizador de uma aplicação de gestão de stocks em componentes [Fedosejev, 2015]. . . .	34
2.30	Exemplo básico da estrutura de um SFC, composto pelo bloco de HTML (<code><template /></code>), pelo bloco de lógica (<code><script /></code>), e pelo bloco de <i>styling</i> (<code><style /></code>)	35
2.31	Excerto de código que exemplifica o encadeamento de métodos possibilitado pelo D3. Neste caso, é criada uma escala linear, mapeando o intervalo de pixels [<code>height, 0</code>] ao domínio de dados descrito no comando <code>domain</code>	38
3.1	Página principal do COVID19INDIA [Covid19India, 2021].	42
3.2	Os gráficos <i>sparkline</i> do COVID19INDIA [Covid19India, 2021]. . . .	43
3.3	O mapa interativo do COVID19INDIA [Covid19India, 2021].	44

3.4	A tabela de dados da página inicial do <i>COVID19INDIA</i> [Covid19India, 2021].	44
3.5	O conjunto de gráficos de linhas do <i>COVID19INDIA</i> [Covid19India, 2021].	45
3.6	Os controlos de tempo dos gráficos de linhas do <i>COVID19INDIA</i> [Covid19India, 2021].	45
3.7	Vista global do <i>BUL</i> [BUL, 2020].	47
3.8	Vista distrital do <i>BUL</i> [BUL, 2020].	47
3.9	Vista municipal do <i>BUL</i> [BUL, 2020].	48
3.10	Quantidade e estado dos trabalhos de construção das redes do <i>BUL</i> [BUL, 2020].	49
3.11	Tecnologias oferecidas pelos operadores do <i>BUL</i> [BUL, 2020].	49
3.12	Horizonte temporal previsto (ou concretizado) da implementação das redes <i>BUL</i> [BUL, 2020].	50
3.13	Captura de ecrã do mapa interativo do <i>UCLA Energy Atlas</i> [Pincetl et al., 2020].	52
3.14	Captura de ecrã da interface de escolha de perfil do <i>UCLA Energy Atlas</i> [Pincetl et al., 2020].	52
5.1	Matriz de risco	64
5.2	<i>Trello</i> - captura de ecrã do quadro <i>Kanban</i> do projeto durante o início do desenvolvimento. O código <i>CS-XX</i> (em que <i>XX</i> é um número qualquer) diz respeito à tarefa. Este código é único e não se repete, fazendo com que a tarefa seja o mais rastreável possível.	65
5.3	Captura de ecrã de um <i>pull request</i> para uma funcionalidade nova, incluindo o título, descrição e URL da tarefa correspondente. Repare-se que o código do <i>pull request</i> é igual ao código da tarefa correspondente.	66
5.4	Captura de ecrã de uma porção dos dados disponíveis. Para além dos atributos que necessitam de atenção, são visíveis as discrepâncias que existem no processamento pós-recolha dos dados - ficheiros diferentes que contêm o mesmo tipo de dados têm estruturas ou organizações diferentes, tornando o seu tratamento mais complexo.	67
5.5	Diagrama ER da base de dados da plataforma - as quatro entidades cruciais e seus atributos.	69
5.6	Captura de ecrã da tabela de localizações, após carregamento do respetivo CSV para o Supabase. Cada entrada nesta tabela caracteriza uma localização, representada pelo seu identificador único, nome, coordenadas geográficas e código hexadecimal de cor para facilitar a distinção dos dados correspondentes a cada localização.	70
5.7	Captura de ecrã da tabela de medições, após carregamento do seu CSV para o Supabase. Cada entrada nesta tabela representa uma medição, caracterizada pelo seu identificador, o identificador da localização onde foi efetuada, datas e horas de início e fim, e o identificador da variável que foi medida.	70

5.8	Captura de ecrã da tabela de variáveis, após carregamento do seu CSV para o Supabase. Cada entrada nesta tabela representa uma variável, que é caracterizada apenas pelo seu identificador e nome.	70
5.9	Captura de ecrã da tabela de valores, após carregamento do seu CSV para o Supabase. Cada entrada nesta tabela é caracterizada pelo seu identificador, o valor registado a uma certa altitude, assim como o identificador da medição onde se insere.	71
5.10	Representação simplificada da arquitetura Cliente-Servidor utilizada no projeto. O cliente React e o <i>backend</i> Supabase são unidos pela API do serviço, através do SDK providenciado.	72
5.11	Captura de ecrã do código necessário para invocar uma função no servidor através de RPC. Neste caso, as instruções na figura chamam a função <i>search_locations_by_string</i> com o argumento <i>search_term</i>	72
5.12	Captura de ecrã da função invocada na figura 5.11. Esta função SQL é executada no lado do servidor após ser chamada pelo cliente, fazendo uma <i>query</i> sobre todas as localizações disponíveis na plataforma e retornando uma cujo nome corresponda ao parâmetro <i>search_term</i> enviado pelo <i>frontend</i>	73
5.13	Captura de ecrã da página de registo.	74
5.14	Captura de ecrã da página de início de sessão.	74
5.15	Captura de ecrã da página de <i>Settings</i> fora do modo de edição - Os campos de <i>input</i> estão todos bloqueados, permitindo que o utilizador consulte as suas definições sem as alterar por engano.	75
5.16	Captura de ecrã da página de <i>Settings</i> no modo de edição - Os campos de <i>input</i> estão ativos, permitindo que o utilizador altere as suas definições.	75
5.17	Seleção de um local de medição pré-existente na plataforma. Ao seleccionar um local, os campos referentes às coordenadas geográficas preenchem-se automaticamente.	76
5.18	Criação de um local de medição novo. Ao digitar um nome de um local que não está guardado na plataforma, é apresentada a opção de criar um novo.	76
5.19	Seleção de uma variável climática pré-existente na plataforma.	77
5.20	Criação de uma variável climática nova. Ao digitar um nome de uma variável que não está guardada na plataforma, é apresentada a opção de criar uma nova.	77
5.21	Carregamento de um ficheiro CSV com os valores correspondentes à localização, variável climática e período de tempo de medição previamente inseridos. Do lado esquerdo, é mostrado o nome do ficheiro carregado e, do lado direito, as suas primeiras linhas.	78
5.22	Exemplo do formato do ficheiro CSV que a plataforma aceita como válido. A primeira coluna corresponde aos valores da variável climática medida, e a segunda coluna contém as altitudes a que a variável foi medida.	78
5.23	Captura de ecrã do <i>dashboard</i> . Esta página apresenta cinco elementos cruciais que interagem entre eles para compôr a ecrã principal da plataforma.	79

5.24	Capturas de ecrã dos dois estados de seleção possíveis para uma localização no mapa.	80
5.25	Captura de ecrã de um utilizador a desenhar a área de interpolação. Quando a área desejada é acabada de desenhar, a camada de interpolação é renderizada sobre o mapa.	81
5.26	Captura de ecrã do resultado da interpolação com IDW. Esta funcionalidade utiliza os valores medidos a uma altitude definida nas localizações selecionadas para calcular e renderizar uma superfície de interpolação. Os valores mais elevados da variável em análise (neste caso a temperatura) são representados com a cor vermelha e os mais baixos com a cor azul.	82
5.27	Captura de ecrã da barra de pesquisa durante uma pesquisa. Os resultados de pesquisa são atualizados enquanto o utilizador escreve. Para além disso, as localizações selecionadas são representadas com um ícone a verde, e as por selecionar são representadas por um ícone cinzento.	83
5.28	Captura de ecrã da barra de pesquisa a seguir a selecionar um dos resultados. Após escolher num dos resultados apresentados, o mapa é centrado na localização clicada.	83
5.29	Captura de ecrã dos filtros de variáveis. Cada variável tem uma cor para que haja a sua fácil distinção no gráfico de linhas que é renderizado no filtro de altitude.	84
5.30	Capturas de ecrã dos dois modos possíveis de filtragem de valores dos filtros de variável e altitude.	85
5.31	Capturas de ecrã dos dois modos possíveis de filtragem de valores do filtro de tempo.	85
5.32	Captura de ecrã do filtro de altitude. Cada variável tem uma cor para que haja a sua fácil distinção no gráfico de linhas que é renderizado. Estas cores correspondem às cores na figura 5.29.	86
5.33	Captura de ecrã da barra de detalhes após seleção de três localidades. Do lado esquerdo, é apresentada uma lista das localidades selecionadas. Do lado direito, o gráfico de pontos com os dados relativos às medições	86
5.34	Captura de ecrã da barra de detalhes com medições diferentes para a mesma localidade. Na imagem, vemos que a localização <i>Casa do Sal</i> tem dois momentos de medição diferentes, o que pode ser confuso para o utilizador. Assim, ao passar com o rato por cima de qualquer uma delas, são reveladas, no canto inferior esquerdo do gráfico, as datas e horas de início e fim dos momentos de medição.	87
B.1	Resultados da primeira pergunta do questionário pós-teste.	110
B.2	Resultados da segunda pergunta do questionário pós-teste.	110
B.3	Resultados da terceira pergunta do questionário pós-teste.	110
B.4	Resultados da quarta pergunta do questionário pós-teste.	111
B.5	Resultados da quinta pergunta do questionário pós-teste.	111
B.6	Resultados da sexta pergunta do questionário pós-teste.	111
B.7	Resultados da sétima pergunta do questionário pós-teste.	111
B.8	Resultados da oitava pergunta do questionário pós-teste.	112

B.9	Resultados da nona pergunta do questionário pós-teste.	112
B.10	Resultados da décima pergunta do questionário pós-teste.	112

Lista de Tabelas

5.1	Requisitos funcionais da configuração da plataforma	58
5.2	Requisitos não-funcionais do desempenho da plataforma	59
5.3	Requisitos não-funcionais da versatilidade da plataforma	60
5.4	Requisitos não-funcionais da usabilidade da plataforma	60
5.5	Descrição do risco #1	61
5.6	Descrição do risco #2	61
5.7	Descrição do risco #3	62
5.8	Descrição do risco #4	62
5.9	Descrição do risco #5	63
5.10	Classificação do impacto de um risco em categorias	63
5.11	Categorias de probabilidade	63
5.12	Resultados da primeira tarefa do teste de usabilidade	91
5.13	Resultados da segunda tarefa do teste de usabilidade	91
5.14	Resultados da terceira tarefa do teste de usabilidade	92
5.15	Resultados da quarta tarefa do teste de usabilidade	93
A.1	Requisitos Funcionais de Autenticação	105
A.2	Requisitos Funcionais das definições de utilizador	105
A.3	Requisitos Funcionais de Visualização	106

Capítulo 1

Introdução

Este capítulo foca-se em providenciar uma visão geral do enquadramento, motivação e principais objetivos que se pretendem atingir com esta dissertação.

1.1 Enquadramento

Atualmente, vivemos num mundo em que a quantidade total de dados criados, capturados, copiados e consumidos globalmente aumenta a cada segundo. Para 2023, prevê-se uma soma de 120 zetabytes de informação em permanente circulação. Durante os próximos dois anos, é expectável que a criação global de dados aumente para mais de 180 zetabytes [Taylor, 2023]. Em 2020, a quantidade de dados criados e replicados atingiu um novo máximo. O crescimento foi superior ao anteriormente esperado devido ao aumento da procura causado pela COVID-19, uma vez que mais pessoas começaram a trabalhar a partir de casa e utilizaram mais frequentemente opções de entretenimento doméstico [Taylor, 2022]. A informação é produzida a um ritmo estonteante, por praticamente todos os domínios da sociedade - indústria, negócio, política, e até os próprios humanos. Com a melhoria dos processos e instrumentos de recolha, a sua qualidade aumenta em conformidade. Na última década, o incremento constante na qualidade dos dispositivos de armazenamento e dos meios para criar e recolher informação influenciaram a maneira como lidamos com os dados em si. No entanto, este progresso tecnológico gera alguns problemas no que toca ao aproveitamento da informação recolhida. Na maioria das vezes, os dados são armazenados sem filtragem, sem segurança, ou até em formatos que não são apropriados para uso posterior. Na maioria das aplicações, os dados em bruto não têm valor em si: o que queremos é extrair o conhecimento subjacente que neles está presente, invisível perante a mera análise dos valores recolhidos. Com isto em mente, temos nas técnicas de visualização analítica (em inglês, *visual analytics* ou apenas VA), uma das possíveis respostas para o problema supramencionado - uma maneira de aliar grandes quantidades de dados à extração de conhecimento, de forma transparente e clara, comunicando visualmente a informação.

Particularmente, a visualização geográfica de dados climáticos refere-se ao pro-

cesso de representação de dados relacionados com o clima num mapa ou noutra representação geográfica. Este tipo de visualização é útil para compreender padrões e tendências dos dados climáticos, assim como para identificar áreas que podem ser particularmente vulneráveis aos impactos de alterações climáticas. A visualização geográfica de dados climáticos pode incluir uma vasta gama de tipos de dados, tais como temperatura, precipitação e subida do nível do mar, e pode ser utilizada para criar mapas interactivos, imagens estáticas, ou mesmo animações. O objectivo desta disciplina é tornar os dados mais acessíveis e compreensíveis para que *decision-makers*, cientistas e o público em geral compreendam melhor o impacto de certas dinâmicas climáticas no nosso planeta. Em aplicações avançadas, a ligação dos inúmeros tipos de dados depende de um variadíssimo leque de visualizações diferentes que são usadas para os representar. Assim, torna-se relativamente trivial perceber que integrar múltiplos modelos de visualização num ambiente único e coerente é uma tarefa trabalhosa.

1.2 Motivação

Atualmente, os laboratórios e os investigadores que os integram continuam a depender demasiado de ferramentas ou técnicas pouco adequadas para extrair conhecimento a partir dos dados que recolhem. Apesar de terem facilidade metodológica - fruto da experiência e formação científica - no que toca à condução de medições precisas, muitos destes profissionais não possuem a disponibilidade ou as capacidades técnicas para processar os dados que daí advêm de forma a serem fáceis de entender. Adicionalmente, é provável que muitos deles não utilizem qualquer tipo de ferramenta de visualização, potenciando o mau aproveitamento ou até corrupção dos dados recolhidos.

Com isto em conta, a visualização analítica é extremamente útil nas situações em que a natureza dos dados é complexa e heterogénea e/ou quando o fenómeno que os gera é desconhecido. O estudo localizado das condições climáticas é um exemplo particularmente interessante deste tipo de cenários. Nestes casos, os dados atmosféricos são geralmente recolhidos a vários níveis de altitude, em múltiplas localizações geográficas e alturas do tempo - geralmente com recurso a *drones* equipados para o efeito. Este método permite capturar e recriar uma imagem topo-climática detalhada, possibilitando a realização de análises aprofundadas em diferentes intervalos de tempo. Assim, dada a complexidade deste tipo de dados e dos fenómenos em questão, há uma especial procura a nível de técnicas de visualização e de interação que confirmem o conhecimento existente e que desvendem novos padrões e tendências nos dados, que, de outra maneira, permaneceriam por desvendar.

Geralmente, para agilizar a compreensão e análise deste tipo de dados, os investigadores desta área científica tiram partido de técnicas de visualização de informação geográfica e ferramentas de visualização analítica. Em particular, trata-se de um conjunto de técnicas de visualização conhecidas (e.g mapas temáticos e/ou gráficos de pontos tridimensionais), que, de forma isolada, são eficazes em situações muito específicas. No entanto, o problema surge quando estas técnicas são

integradas numa ferramenta analítica única. Cada uma das técnicas tem as suas especificidades (conceptuais e de implementação) que dificultam a integração do código e especialmente a partilha dos dados. Assim, há uma especial motivação para o desenvolvimento de soluções que harmonizem várias técnicas de visualização, de forma simples e integrada.

No panorama geral do desenvolvimento para a *web*, e em específico, no desenvolvimento de interfaces de utilizador, podemos facilmente identificar uma tendência crescente nos últimos anos: a modularidade. Com o desenvolvimento e aumento exponencial do uso de tecnologias como o React.js, Vue.js, e Angular.js, observa-se uma contínua evolução em direção à compartimentalização e à separação de preocupações. Esta abordagem apoia radicalmente a reutilização de código, levando à sua mais fácil manutenção e legibilidade. Curiosamente, o paradigma atual da visualização de dados e, no geral, do desenvolvimento de plataformas de visualização analítica carece deste tipo de abordagens, especialmente no que toca a visualizações avançadas. Desta forma, foi identificado um nicho tecnológico nas áreas de visualização de dados, que tira especial partido das vantagens da modularidade.

O desenvolvimento de uma plataforma modular de visualização climática permitirá a indivíduos com formação científica mudar o seu foco para simplesmente analisar os dados que recolheram, abstraindo todo o processo de tratamento de dados e implementação de visualizações apropriadas. Com recurso a uma ferramenta deste tipo, prevê-se que estes profissionais gastem menos tempo em tarefas fora do âmbito da investigação de fenómenos climáticos localizados. Assim, pode ser apresentado um argumento válido que defenda a criação e desenvolvimento de uma plataforma que agregue múltiplos tipos modulares de visualização climática, que remova a componente de processamento e tratamento de dados, e que, em última instância, providencie aos investigadores uma compreensão mais profunda das problemáticas em questão.

1.3 Âmbito e Objetivos

O principal objetivo deste trabalho consiste em desenvolver uma plataforma de visualização de dados climáticos que permita analisar os dados recolhidos de uma maneira fácil, intuitiva e rápida, para indivíduos ou equipas com formação científica na área. Para além disto, esta plataforma precisa de possibilitar a confirmação de fenómenos climáticos localizados conhecidos, assim como a exploração e descoberta de novos fenómenos. Desta forma, acelera-se o processo de estudo deste tipo de acontecimentos, de forma independente e específico a cada localidade estudada.

O desenvolvimento de uma plataforma deste tipo, e todas as suas práticas subjacentes, nasce da deteção de um rifte entre a necessidade de juntar vários tipos de visualização diferentes (p.e. geográfica e analítica) e a falta de soluções que alavanquem o conceito de modularidade em contexto *web*. Para levar a cabo esta ideia, é importante começar por conceptualizar o uso que será dado à plataforma, dando resposta às necessidades posteriormente identificadas, através da criação

de componentes de visualização avançada, que actuarão como peças para compor a interface da plataforma - em particular, técnicas de visualização que tiram partido da representação das estruturas geo-temporais. O objectivo será transformar os modelos de visualização existentes em componentes *web* hierárquicos. Numa segunda fase, será desenvolvido o sistema de visualização que integra diversos componentes de visualização e os modelos avançados específicos ao problema de visualização dos dados atmosféricos. Finalmente, os resultados obtidos serão validados através de testes de utilizador. O vídeo demonstrativo da plataforma pode ser acedido <https://youtu.be/eQbIrndDvtM>.

1.4 Estrutura do Documento

Esta secção pretende identificar e descrever o conteúdo de cada um dos capítulos apresentados no documento.

1. **Introdução** - É o primeiro capítulo do documento que tem como principal objetivo orientar os leitores quanto ao tema e as intenções da dissertação;
2. **Estado da Arte** - Enquadra o estado da produção de conhecimento científico no que diz respeito ao tema proposto, enfatizando as diversas técnicas utilizadas nas várias áreas dentro da visualização;
3. **Trabalho Relacionado** - Análise do mercado existente e a forma como a implementação de uma plataforma de visualização contribui para a descoberta de novo conhecimento e confirmação do conhecido;
4. **Metodologia** - Mapeamento de esforços e objetivos pretendidos com este estudo, bem como a estratégia apontada para a sua concretização tendo em conta todas as componentes que envolve, desde a avaliação de necessidades ao desenho e implementação da plataforma e suas componentes, assim como a validação do sistema com base na sua utilização;
5. **Trabalho Prático** - Descrição e análise da implementação das quatro fases acima referidas (avaliação, design, desenvolvimento e validação), propondo a forma como este pode solucionar as problemáticas identificadas;
6. **Conclusão** - Este capítulo consiste em analisar e refletir sobre os resultados obtidos, apresentando uma síntese dos mesmos e ainda alguns aspetos de melhoria e dificuldades sentidas;
7. **Referências** - Elicitação das fontes consultadas e recursos utilizados ao longo do documento;
8. **Anexos** - Informações e recursos complementares ao documento, incluindo tabelas de requisitos.

Capítulo 2

Estado da Arte

Este capítulo foca-se em recolher, sintetizar e analisar o estado atual dos pilares desta dissertação. São introduzidos conceitos básicos do campo da visualização no geral. Aprofundam-se várias noções que compõem a visualização geográfica e analítica, e as técnicas que permitem ao utilizador interagir com o software de visualização. Para além disso, são discutidas algumas tecnologias proeminentes no panorama geral do desenvolvimento *web*, que, apesar de não serem o foco do projeto, são inerentemente importantes.

2.1 Visualização Geográfica

Nos últimos anos, a visualização de dados geoespaciais tem sofrido desenvolvimentos dramáticos e importantes. Para além de mapas estáticos, existem ambientes virtuais interactivos que podem ser utilizados para explorar e apresentar dados geoespaciais de forma dinâmica. Além disso, a Internet é um meio proeminente para divulgar dados e mapas geoespaciais.

Com o advento do uso de dispositivos de computação ubíqua (p.e *smartphones* e *drones*), os dados geo-referenciados são recolhidos em grande quantidade e com elevada velocidade. Assim, as técnicas de visualização geográfica - a aplicação da visualização para apoiar a exploração de dados geo-espaciais -, tornaram-se mandatórias para conseguir destilar a informação presente nestes dados e deles extrair conhecimento [Kraak, 2003].

Um sub-conjunto das variadas técnicas de visualização geográfica consiste nos mapas temáticos. Os mapas temáticos são um tipo de visualização geográfica que é utilizado para exibir dados e informações que são organizados por um tema ou assunto específico. Os dados são tipicamente representados utilizando símbolos, cores e/ou outros elementos visuais, que são colocados num mapa de acordo com a localização dos dados. Os mapas temáticos podem ser utilizados para exibir uma grande variedade de informações, tais como densidade populacional, actividade económica ou recursos naturais. Para além de vários tipos de mapas temáticos - mapas de pontos, cloropléticos, de superfície e de fluxo -, nesta secção são discutidos também os símbolos que os compõem, dada a sua relevância para

este projeto.

2.1.1 Símbolos Cartográficos

Em 1967, Jaques Bertin estabeleceu a teoria de cartografia mais indispensável para a transcrição de informação geográfica [Bertin, 1983; Koch, 2000]: um sistema de variáveis visuais que descrevem a percepção das diferenças nos símbolos cartográficos que são utilizados para representar fenómenos geográficos [Slocum et al., 2008]. Assim, os cartógrafos distinguem quatro símbolos essenciais [Robinson, 1960; Sandvik, 2008; Slocum et al., 2008]:

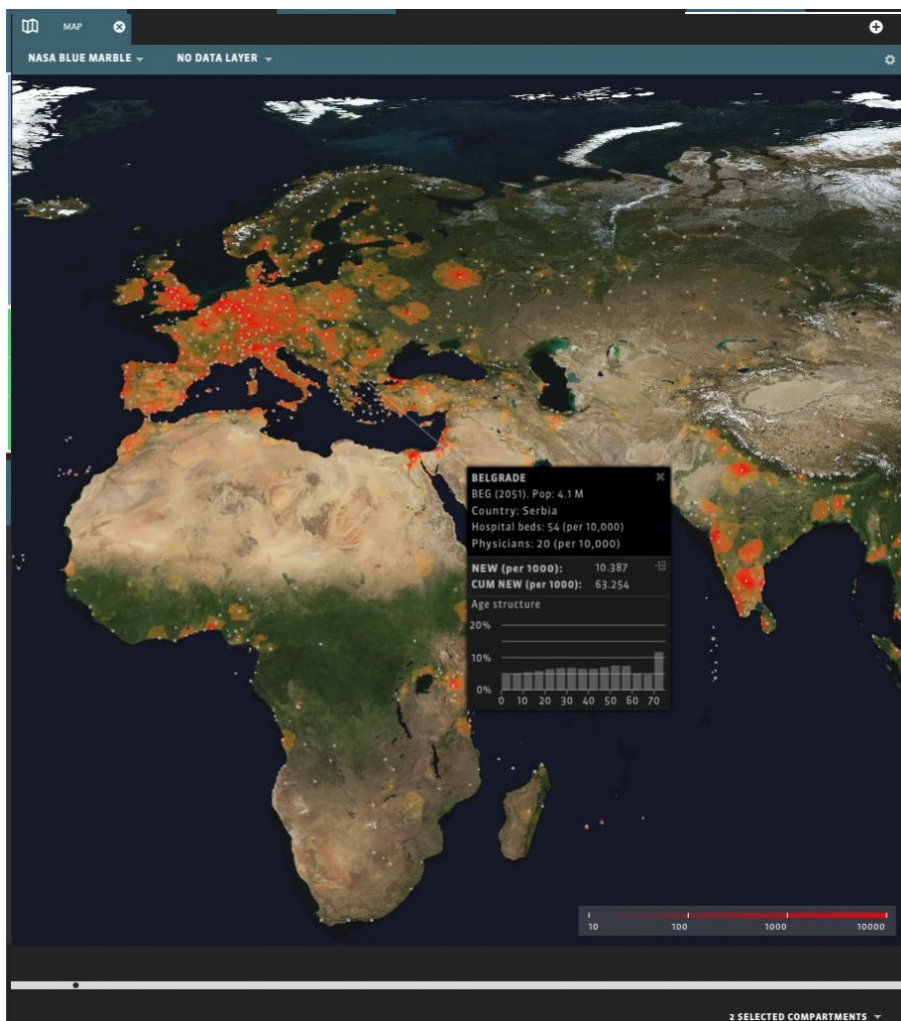


Figura 2.1: Captura de ecrã de uma porção da interface do GLEAMViz, uma plataforma de controlo e análise da propagação de doenças. As ocorrências de uma determinada doença são representadas por pontos vermelhos sobre o mapa [GLEAM, 2022].

- **Ponto** – Este símbolo refere-se a um determinado local no espaço. Paralelamente, o ponto não tem dimensão teórica e só é visível através da marca que o renderiza. Assim, existem duas principais aplicações no contexto da cartografia [MacEachren, 1979]:

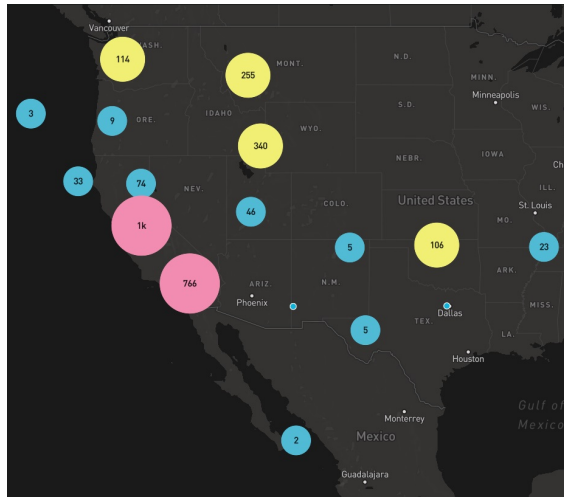


Figura 2.2: Mapa de símbolos em React-Mapbox. Os pontos são agregados em *clusters* conforme o nível de zoom. Para além disso, as cores dos *clusters* são modificadas conforme o número de elementos que contêm - a cor rosa corresponde a um maior número de elementos, e a cor azul corresponde a um menor número de elementos [vis.gl, 2017].

- Os fenómenos geográficos a serem mapeados estão localizados num local. A figura 2.1 exemplifica a utilização deste símbolo. Nesta figura, é apresentada uma captura de ecrã da plataforma GLEAMViz [GLEAM, 2022]. O GLEAM - *GLobal Epidemic And Mobility model* -, combina dados reais sobre populações e mobilidade humana, com modelos estocásticos de transmissão de doenças para fornecer poder analítico e de previsão. É uma ferramenta de apoio ao desenvolvimento de estratégias de intervenção que minimizem o impacto de epidemias potencialmente devastadoras. Na imagem, podemos observar que o ponto é utilizado para codificar um local geográfico - neste caso, os locais de incidência de uma determinada doença.
- Os fenómenos geográficos a serem mapeados são agregados numa dada localização. O exemplo da figura 2.2 ilustra uma implementação deste conceito em *React-Mapbox*, uma *package* de *Node.js* que permite tratar mapas como elementos da *DOM* do *browser*. Nele, os pontos a serem representados são agregados num *cluster* conforme o nível de *zoom* aos mesmos - quanto menor o *zoom*, maior o agregamento e menor a resolução dos pontos individuais [vis.gl, 2017].
- **Linha** – A linha é uma marca visual que liga dois pontos no espaço, tem um comprimento definido, e não apresenta qualquer área mensurável. A linha é utilizada para indicar conectividade ou fluxo, valores iguais ao longo de uma linha e fronteiras entre áreas diferentes [MacEachren, 1979]. O símbolo é diferenciado com base na sua forma (por exemplo, linha sólida versus linha pontilhada), cor e largura. Humboldt foi um dos primeiros a traçar linhas a passar pelos lugares com a mesma temperatura média anual no mapa, a que chamou 'isotermas' - (da palavra grega para 'igual' - *isos* e 'calor' - *thermos*). O mapa foi publicado em Paris, em 1817 [Robinson and

Wallis, 1967] e está representado na figura 2.3. Embora este exemplo esteja longe de ser a primeira utilização de linhas numa visualização, é um que a ilustra de forma bastante elegante.

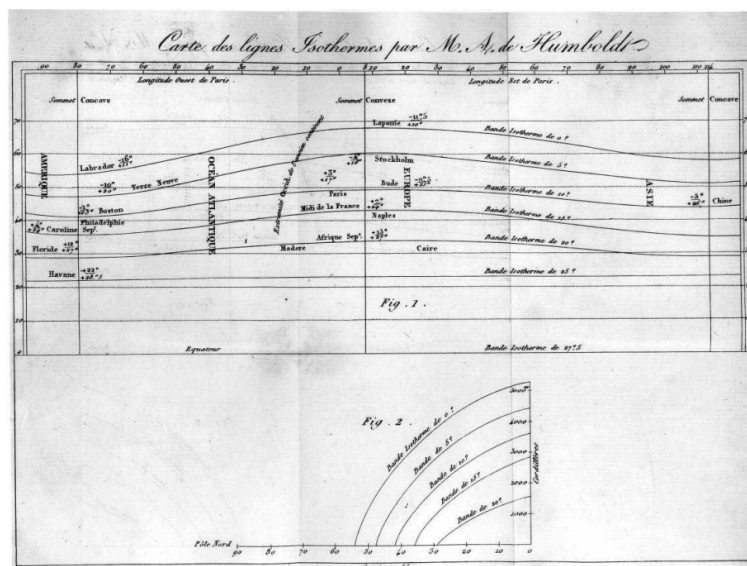


Figura 2.3: Linhas isotérmicas de Humboldt. Neste exemplo, as linhas são utilizadas para ligar os locais com a mesma temperatura média anual [Robinson and Wallis, 1967].

- **Área** – Os símbolos de área são utilizados para atribuir uma característica ou valor a uma área inteira num mapa. As variáveis visuais utilizadas para os símbolos de área são cor, textura e altura da perspectiva [Slocum et al., 2008]. No trabalho de Loomis, são descritas duas tempestades que foram vividas nos Estados Unidos, no mês de Fevereiro de 1842 [Loomis, 1845]. Nesta investigação, o autor tem acesso a dados meteorológicos de outros investigadores interessados, resultando no mapa da figura 2.4. Nele, as regiões onde o céu estava sem nuvens são de cor azul; aquelas onde o céu estava totalmente encoberto, mas sem chuva ou neve, são de cor castanha; a queda de neve é indicada pela cor verde, e a chuva pelo amarelo. O autor utiliza o conceito de área para transmitir diferentes estados da meteorologia durante as tempestades estudadas.
- **Volume** – No contexto da visualização geográfica, o volume diz respeito à representação tridimensional de dados ou informação num espaço geográfico. Os símbolos de volume podem ser classificados em duas subcategorias: 2½-D e verdadeiro 3-D [Slocum et al., 2008]. O primeiro pode ser pensado como uma superfície, na qual uma localização geográfica é definida por pares de coordenadas (x,y) e o valor do fenómeno é a altura acima de um ponto zero [Sandvik, 2008]. O projeto LAEI-NO2, desenvolvido pelo grupo Parallel em 2020 e representado na figura 2.5, vemos a distribuição da concentração do dióxido de nitrogénio, NO_2 , na área da Grande Londres [Parallel, 2020]. Este projeto usa o volume para representar a concentração de NO_2 , consoante o local geográfico em questão. Dada a complexidade visual da representação escolhida, a solução utiliza também um espetro de

cores (presente no canto superior esquerdo) para codificar a concentração - a cor azul codifica a menor concentração presente nos dados, e a cor vermelha codifica a maior.

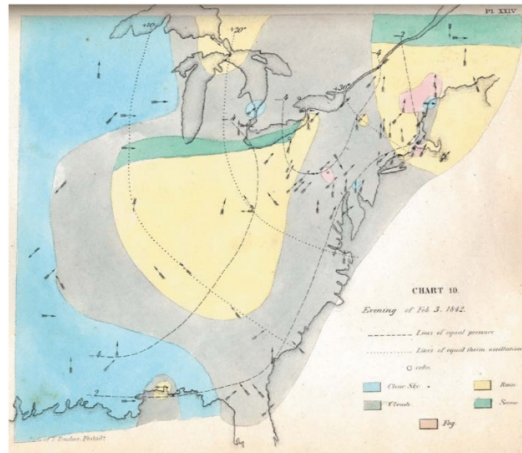


Figura 2.4: Estudo de Loomis sobre duas tempestades que ocorreram em Fevereiro de 1842. A área é utilizada para comunicar diferentes estados meteorológicos [Loomis, 1845].

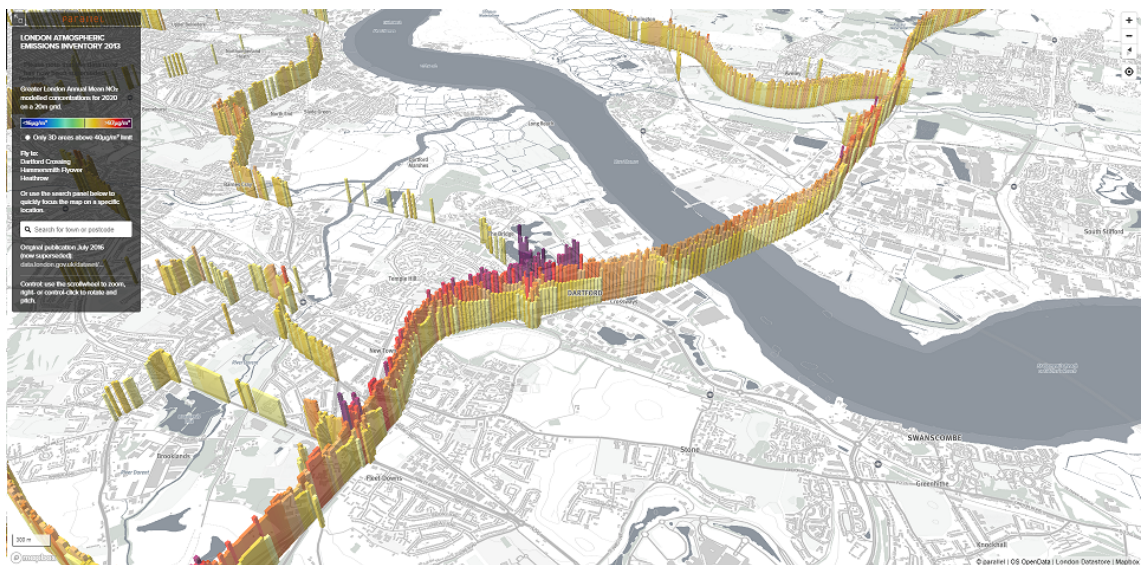


Figura 2.5: Captura de ecrã da interface do LAEI-NO2. Neste exemplo, o volume é utilizado para representar a concentração de dióxido de nitrogénio [Parallel, 2020]

2.1.2 Mapas de Pontos

Os mapas de pontos são um tipo de visualização que retrata a distribuição espacial de dados geográficos discretos [Roth, 2017]. Em geral, uma colecção de pontos indica a presença de certos objectos numa região geográfica. Os pontos

podem ainda ser enriquecidos com mais informação, dotando-os de cores ou outras indicações visuais para exibir valores de dados adicionais, tanto categóricos como numéricos. Em muitos casos, os pontos no mapa não devem ser contados individualmente, mas interpretados colectivamente. Por exemplo, pontos mais próximos parecem densos e numerosos quando coalescem visualmente [Kimerling, 2009]. Por outro lado, parecem esparsos quando isolados, e tendem a sugerir que estamos perante *outliers*. Consequentemente, padrões de densidade variáveis transmitem a forma como os dados são distribuídos por uma dada região geográfica [Chua and Moere, 2017].

Um famoso exemplo deste tipo de mapas é aquele elaborado por Dr. Snow, em 1854, que apresentamos na figura 2.6. Este mapa retrata a localização geográfica da incidência de mortes causadas por cólera em Londres. Com este mapa, demonstrou-se que havia uma correlação relevante entre as localizações das mortes, e as localizações de bombas de água infetadas [Tennekes, 2018]. Este trabalho estabeleceu o precedente para que este tipo de mapas fosse usado de forma mais recorrente em problemas relacionados com a disseminação de doenças.

Efetivamente, a figura 2.7 é um mapeamento espacial da febre Q na Holanda [Soetens et al., 2017], onde cada ponto representa um caso da doença mencionada.

Seguindo a mesma lógica, o último exemplo, retratado na figura 2.8, utiliza esta técnica como uma ferramenta epidemiológica para mapear as infeções causadas por *Schistosoma mansoni* no Brasil [Barreto, 1993], muito à semelhança da metodologia utilizada por Snow.



Figura 2.6: Mapa de mortes causadas pela bactéria da cólera em Londres, por Snow. Este estudo mapeia as localização geográficas - representadas por pontos - das incidências da incidência de cólera em Londres [Tennekes, 2018].

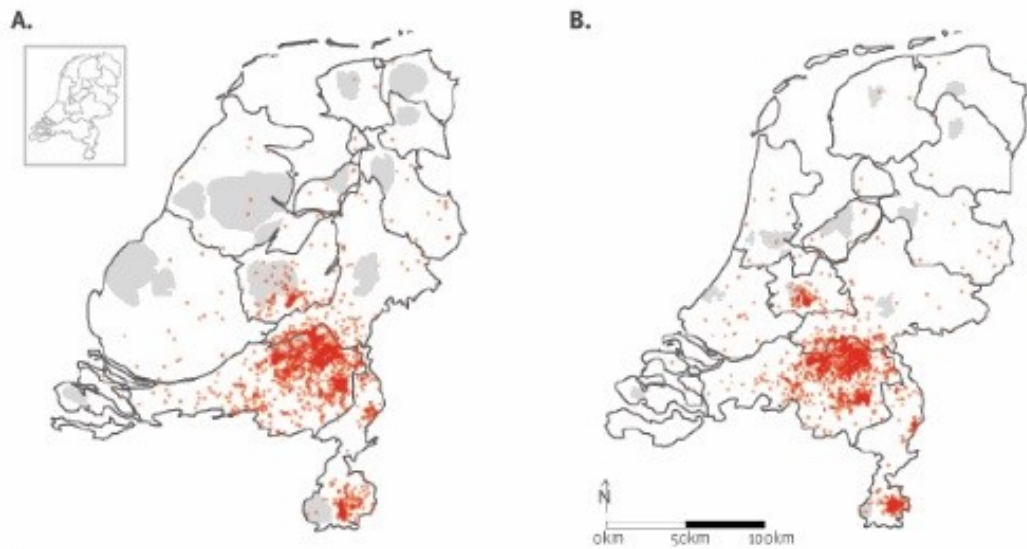


Figura 2.7: Mapa das ocorrências da febre Q na Holanda. À semelhança do mapa do Dr. Snow, as localizações geográficas de ocorrências da doença são representadas por pontos vermelhos sobre um mapa da Holanda [Soetens et al., 2017].



Figura 2.8: Mapeamento das infecções por *Schistosoma mansoni* no Brasil [Barreto, 1993].

2.1.3 Mapas Coropléticos

Os mapas coropléticos são utilizados para representar dados utilizando símbolos de área. Com esta técnica, cada área, que é normalmente uma unidade administrativa, é preenchida com um valor de cor, saturação ou textura correspondente, dependendo dos dados que se pretendem representar. A luminosidade é normalmente utilizada para representar dados categóricos [Brewer and Pickle, 2002]. Uma das maiores desvantagens desta técnica é que o tamanho da área geográfica influencia a percepção das quantidades representadas. Consequentemente, são utilizados valores normalizados em vez de dados absolutos [Dent, 1999; Polisciuc, 2021].

Um exemplo histórico é o mapa do analfabetismo em França, produzido por Charles Dupin em 1826 [Dupin, 1826], também conhecido como o primeiro mapa estatístico moderno. Este mapa retrata a distribuição e intensidade do analfabetismo em França por meio do sombreamento (figura 2.9)

Na figura 2.10, está representado o número de prostitutas em cada um dos 48 *quartiers* de Paris, que dividiam a cidade em 12 *arrondissements*, produzido por Parent-Duchâtelet em 1836 [Parent-Duchatelet, 1836]. Este mapa utiliza sombreamento mais escuro para representar um maior número de prostitutas - presumivelmente baseado no número de detenções feitas pela polícia. O terceiro exemplo, exposto na figura 2.11, ilustra o progresso da implementação de redes de telecomunicação em Itália [BUL, 2020]. Quanto mais intenso for o tom de azul, maior o progresso. Este mapa, assim como o software que o integra, é explicado em maior detalhe na secção 3.2.

2.1.4 Mapas de Superfície

Este tipo de mapa está dividido em duas classes: o mapa isorítmico (por exemplo, mapa topográfico e de temperatura) e o mapa isotérmico (por exemplo, densidade populacional). No primeiro caso, cada isolinha representa igual valor de dados numa distribuição contínua, ou seja, cada isolinha liga pontos geográficos com valores reais iguais. Por outro lado, no mapa isoplético, as linhas do mapa representam dados derivados dos valores em locais especificados, assumindo uma dada função de interpolação.

Os mapas de isopletas diferem dos mapas coropléticos na medida em que os dados não são agrupados numa unidade pré-definida como um distrito da cidade. Estes mapas podem assumir duas formas: Linhas de igual valor são desenhadas de tal forma que todos os valores de um lado da linha são superiores ao um dado valor e todos os valores do outro lado são inferiores; ou gamas de valores semelhantes são preenchidas com cores ou padrões semelhantes. Este tipo de mapa é ideal para mostrar variações graduais no espaço e evita as mudanças abruptas que as linhas de fronteira produzem nos mapas coropléticos. A temperatura, por exemplo, é uma medida contínua, mas não muda abruptamente em nenhum ponto (como a densidade populacional pode fazer ao atravessar para outra zona de recenseamento), logo é um fenómeno que deve ser mapeado usando isopon-

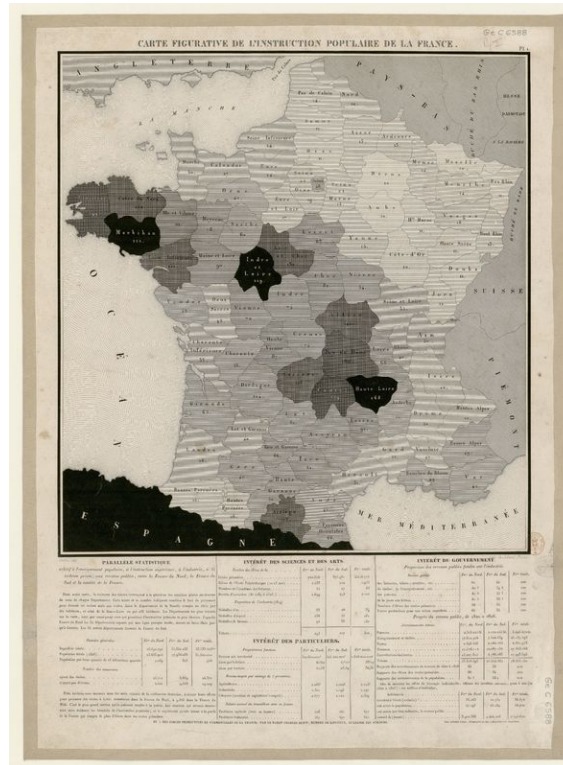


Figura 2.9: Mapa do analfabetismo em França. Neste mapa, as várias tonalidades de cinza representam o grau de analfabetismo (uma variável categórica) nas unidades administrativas do país - tons mais escuros representam maior analfabetismo [Dupin, 1826].

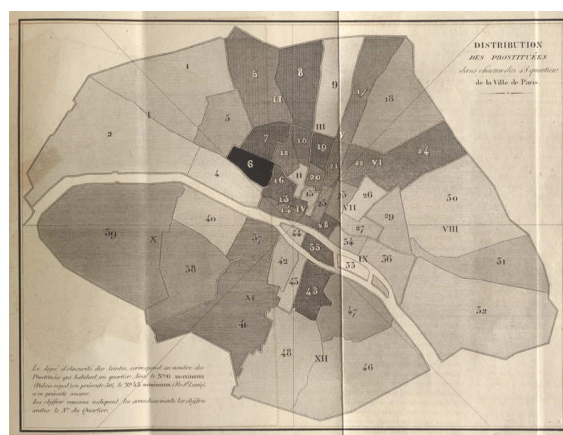


Figura 2.10: Mapa da prostituição em França [Parent-Duchatelet, 1836].

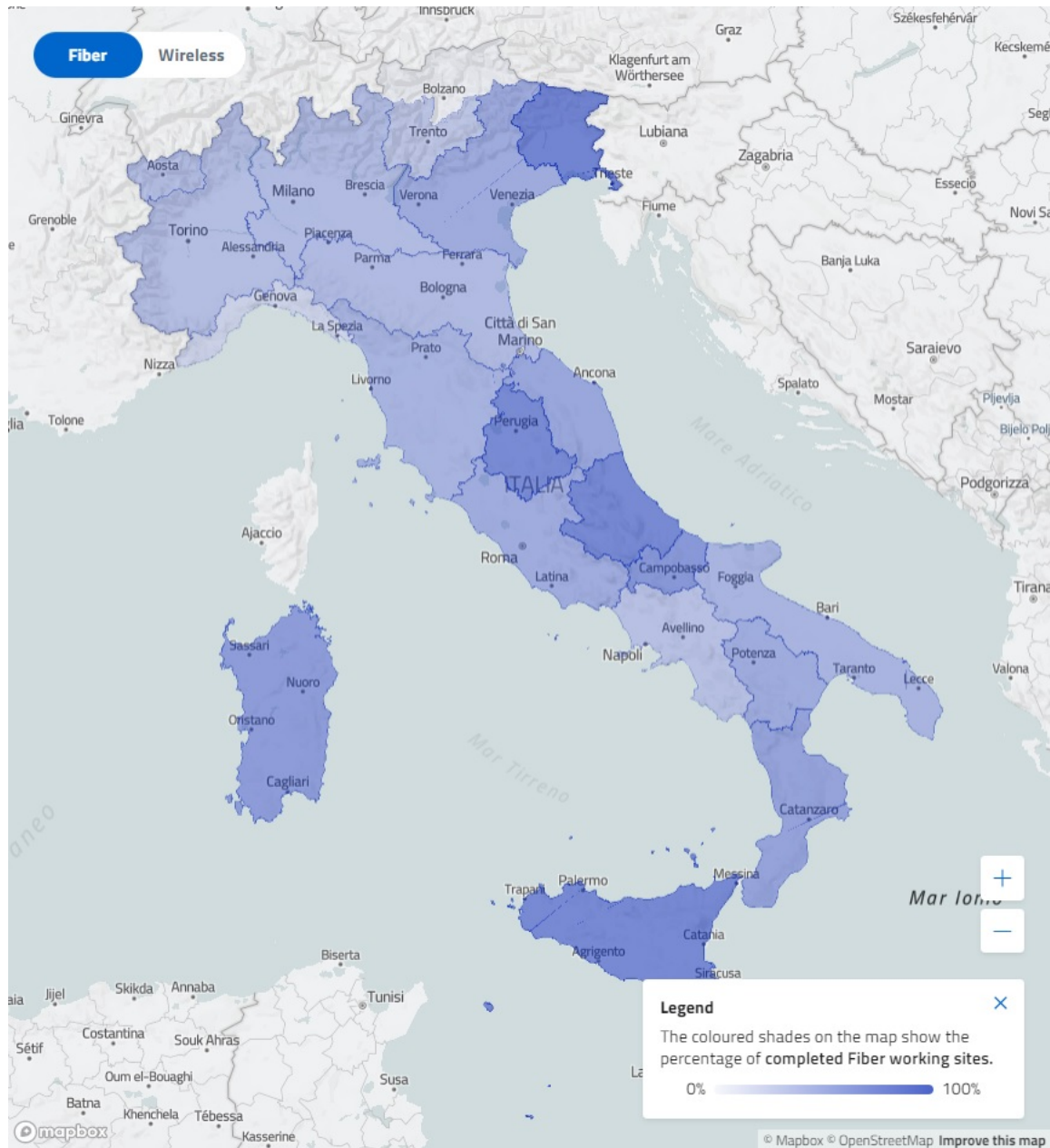


Figura 2.11: Mapa do progresso da implementação de redes wireless e fibra em Itália. As tonalidades coloridas no mapa indicam a percentagem de locais de instalação de fibra que terminaram o seu trabalho [BUL, 2020].

tos. Este argumento é também válido para explicar o facto de o relevo ser representado da mesma maneira em mapas cartográficos [Polisciuc, 2021]. Baseando-se nas descobertas de Humboldt (secção 2.1.1 - Linha), Woodbridge publica a sua própria carta isotérmica mundial em 1823, na qual descreve a relação entre a temperatura média anual e locais geográficos, e ilustrou as regiões climáticas a cores para indicar a sua aptidão para produtos agrícolas, ao mesmo tempo que indicava ainda mais as suas gamas potenciais entre latitudes [Woodbridge, 1823]. O gráfico, intitulado "Isothermal Chart, or View of Climates & Production, Drawn from the Accounts of Humboldt & Others", está representado na figura 2.12.

Considerando exemplos mais recentes, a figura 2.13. ilustra a precipitação a este de Espanha, em Junho de 2000 - zonas com a mesma tonalidade de azul codificam a mesma pluviosidade [Centre, 2022]. Na mesma lógica, o trabalho de Pons, Antoni e Rull - apresentado na figura 2.14 -, ilustra a distribuição do alojamento turístico nas ilhas Baleares, em Espanha, em 1985, 1996 e 2010 [Pons et al., 2014]. Com esta técnica, pode-se distinguir facilmente os *hotspots* das habitações para turismo.



Figura 2.12: Mapa de linhas isotérmicas de Woodbridge [Woodbridge, 1823].

2.1.5 Outros Mapas

Esta porção do relatório foca-se brevemente noutros tipos de mapas temáticos que não são tão relevantes no contexto do projeto pela natureza dos dados envolvidos e tipo de análise que proporcionam. No entanto, foram incluídos por fornecerem um contexto importante acerca da variedade enorme que existe no mundo da visualização.

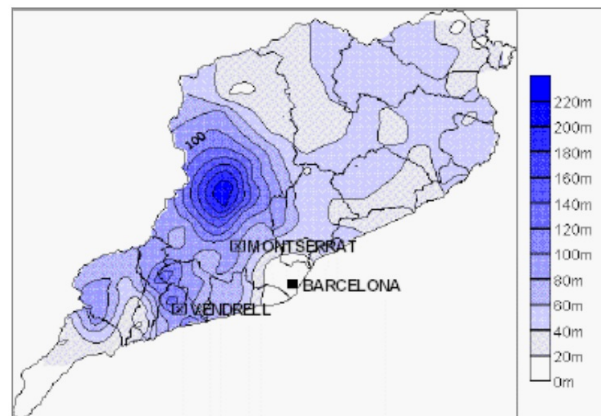


Figura 2.13: Precipitação na região oriental de Espanha, a Junho de 2000 [Centre, 2022].

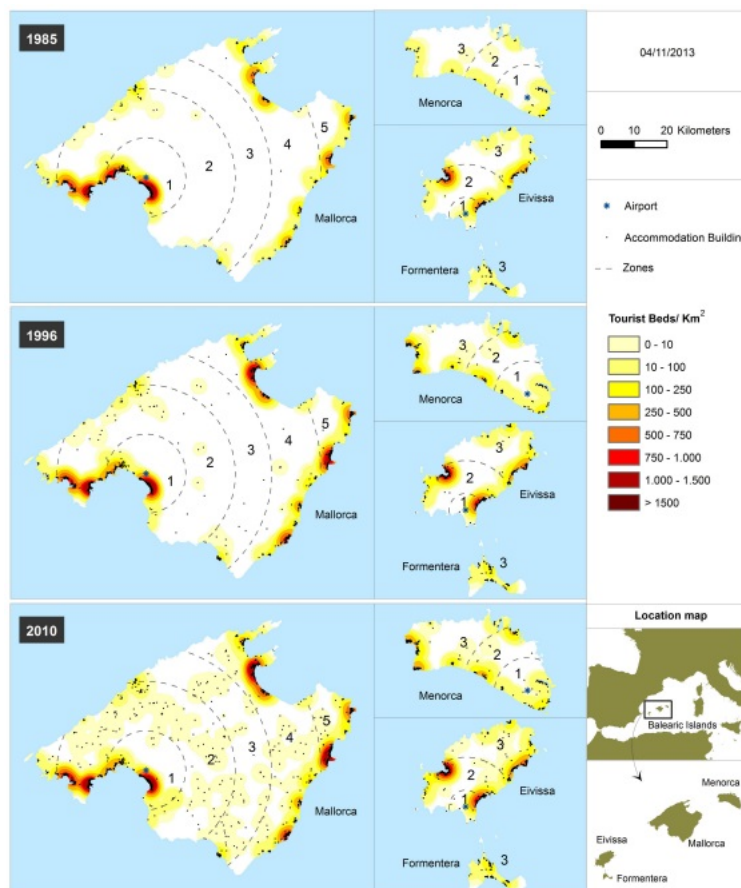


Figura 2.14: Alojamento turístico nas ilhas Baleares em 1985, 1996, e 2010 [Pons et al., 2014].

Mapas de Fluxo/Rede

Os mapas de fluxo e de rede retratam fenômenos lineares que envolvem, na maioria das vezes, movimento e ligação entre pontos dois pontos: uma origem e um destino. Alguns exemplos são mapas que descrevem o fluxo de migrações no mundo ou uma rede de amigos no Facebook. A maioria dos mapas codifica dados multivariados utilizando os atributos visuais de largura de linha, matiz de cor e propriedades espaciais, sendo estas últimas fornecidas pela localização geográfica dos dados [Meirelles, 2013]. Um exemplo histórico é o trabalho de Charles Minard, de 1864 [Minard, 1869]. Nele, descreve-se a exportação de vinhos com origem francesa, onde a largura do elo origem-destino codifica as quantidades exportadas [Phan et al., 2005] (figura 2.15)

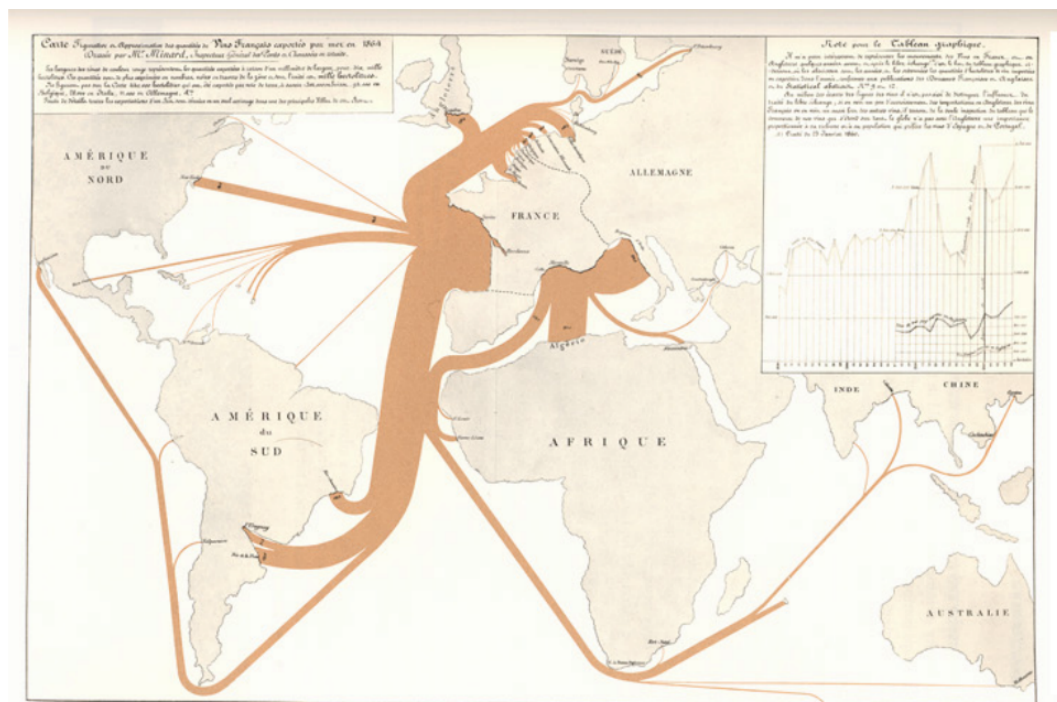


Figura 2.15: Exportações de vinho francesas em 1864, por Minard [Minard, 1869].

Mapas de Símbolos Proporcionais

Os mapas de símbolos proporcionais utilizam o tamanho do símbolo para representar proporcionalmente as magnitudes dos dados discretos temáticos. O tamanho é proporcional às quantidades representadas, mas não depende da área geográfica sobre a qual se encontra. Esta característica evita que se confunda a área geográfica com valores de dados, como no caso dos mapas cloropléticos (secção 2.1.3.). Há duas variáveis principais a considerar na concepção de um mapa de símbolos proporcionais: a forma do símbolo e o seu tamanho. A forma das marcas pode variar, e a forma mais comum é o círculo, embora vejamos barras retangulares, bem como triângulos a serem utilizados. Houve tentativas de símbolos tridimensionais, em que a escala é feita ao cubo e não à raiz quadrada.

Esta abordagem apresentava uma desvantagem grande - se a percepção de área já é difícil a duas dimensões, esse problema exacerba-se quando passamos para uma dimensão superior. Para além disso, é comum ver glifos a serem utilizados - possibilitam enriquecer o mapa com mais informação (i.e mais variáveis) do que um símbolo [Meirelles, 2013]. O mapa de 1858 de Charles Minard é o primeiro exemplo conhecido desta técnica [Friendly, 2002]. A figura 2.16. é uma representação gráfica da quantidade e das proporções de carne produzida por diferentes regiões em França. Cada círculo está dividido em segmentos, onde cada segmento representa uma região diferente e o seu tamanho representa a quantidade de carne produzida nessa região. Esta visualização ajuda a mostrar claramente as contribuições relativas de cada região para a produção de carne em França.

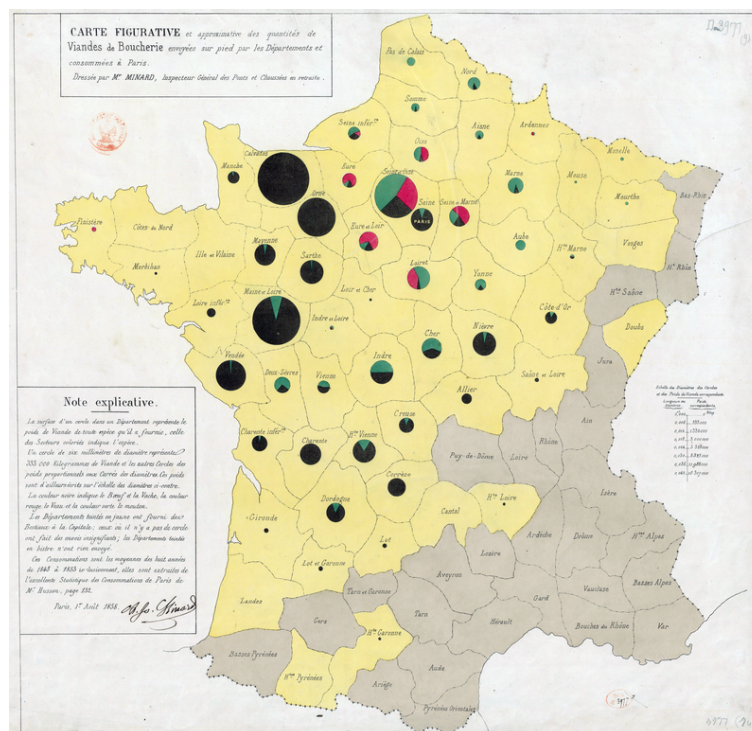


Figura 2.16: Produção de carne em França [Friendly, 2002].

Cartogramas

Ao contrário dos mapas tradicionais, em que o espaço é utilizado para representar o espaço, os cartogramas distorcem o formato das regiões geográficas para codificar outra variável qualquer. Este tipo de mapas utiliza as suas variáveis espaciais para representar os dados da população de acordo com um variável temática. Para permitir a identificação dos espaços geográficos conhecidos, a maioria dos cartogramas de área utilizam algoritmos que mantêm o espaço transformado o mais próximo possível do espaço geográfico [Meirelles, 2013]. O cartograma apresentado na figura 2.17 mostra a distribuição de votos para os dois principais candidatos nas eleições presidenciais dos EUA em 2020 [WorldMapper, 2020]. Cada cor codifica o candidato (Joe Biden, a azul, ou Donald Trump, a vermelho) que recebeu a maior percentagem de votos em cada condado. O próprio cartograma mostra uma projecção de população igual (canto superior direito) onde

cada quadrícula do mapa é redimensionada de acordo com o número total de pessoas que aí vivem.

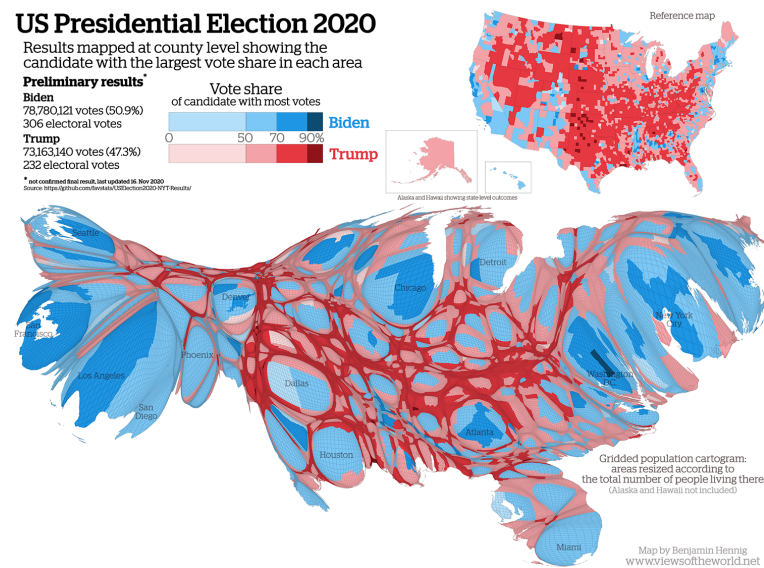


Figura 2.17: Cartograma das eleições presidenciais de 2020. Neste exemplo, a área de cada estado é distorcida de acordo com o seu número de habitantes [World-Mapper, 2020].

2.2 Visualização Analítica

A visualização analítica (VA) combina técnicas de análise automatizada com visualizações interactivas para uma compreensão, raciocínio e tomada de decisão eficazes com base em conjuntos de dados muito grandes e complexos. O objectivo da visualização analítica é a criação de ferramentas e técnicas que permitam que o utilizador sintetize a informação e atinja uma maior e melhor compreensão dos dados. Para além disso, a VA permite detectar o que já conhecemos e descobrir o que não conhecemos, assim como fornecer argumentos fundamentados para justificar uma certa ação consequente - do domínio da economia ao do desporto [Brehmer et al., 2014; Munzner, 2014].

A visualização analítica é relativamente recente no vasto panorama da visualização. A definição inicial e provavelmente a mais proeminente foi fornecida por Cook e Thomas [Cook and Thomas, 2005] - "A visualização analítica é a ciência do raciocínio analítico através de interfaces visuais interactivas"[Polisciuc, 2021]. Assim, este ramo pode antes ser visto como uma abordagem integral à tomada de decisões, combinando a visualização, factores humanos e análise de dados.

A visualização actua como uma plataforma de comunicação das outras peças que compõem a visualização analítica: gestão de dados, *data mining*, análise de dados espacio-temporais e percepção/conhecimento humano (figura 2.18) [Keim and Zhang, 2011]. Tendo em conta todas estas disciplinas e peças envolvidas, a solução que a visualização analítica oferece é permitir que o utilizador possa fazer

parte do processo de aquisição e geração de conhecimento: os dados podem ser manipulados de forma interactiva para obter uma visão mais clara e isenta de preconceitos - tanto sobre os dados como sobre a representação propriamente dita.

O esquema apresentado na figura 2.19 estrutura um processo geral de descoberta de conhecimento ancorado na visualização analítica [Keim et al., 2008]. Para além de representar o *loop* cognitivo do qual o utilizador faz parte, este esquema ilustra também a avaliação da visualização em termos de eficiência ou conhecimento adquirido. Uma escolha para uma representação inicial e interações adequadas pode ser feita após a aplicação de diferentes técnicas estatísticas e matemáticas. O processo entra então num ciclo onde o utilizador pode adquirir conhecimentos sobre os dados, conduzindo idealmente o sistema para técnicas analíticas mais focadas e adequadas. Interagindo sobre a representação visual, o utilizador ganhará uma melhor compreensão da própria visualização e, em última análise, confirmando hipóteses construídas a partir de iterações anteriores [Keim et al., 2008; Keim, 2002].

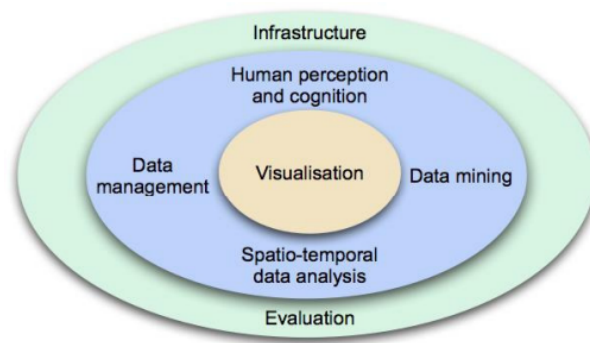


Figura 2.18: Disciplinas que compõem a visualização analítica [Keim and Zhang, 2011].

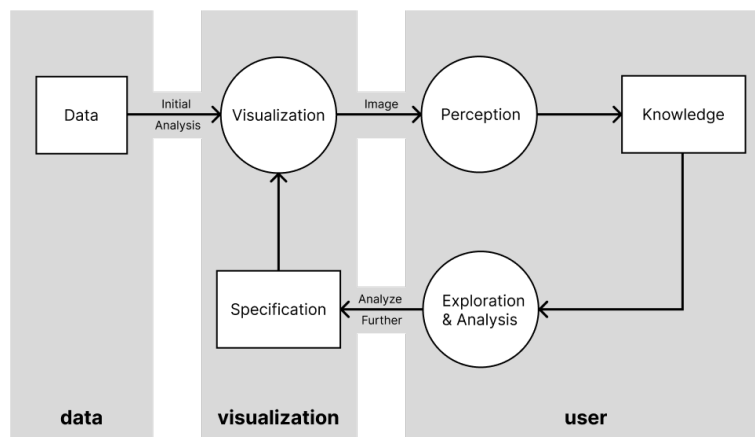


Figura 2.19: O ciclo de aquisição e geração de conhecimento [Keim et al., 2008].

2.2.1 Dashboards

Os *dashboards* oferecem uma perspectiva analítica sobre conjuntos de dados grandes e complexos. Estas ferramentas combinam representações visuais e outros elementos gráficos para fornecer camadas de abstracção e simplificação para grandes quantidades de dados relacionados, de modo a que, quem o consome, tenha uma perspectiva holística da informação mais importante ou relevante, no menor tempo possível. A capacidade que os *dashboards* têm de fornecer uma visão geral levou à sua utilização generalizada nos mais variados domínios do conhecimento, tais como negócios, enfermagem e hospitais, saúde pública, análise de aprendizagem, análise urbana, análise pessoal, energia e muito mais. Estes exemplos, concebidos principalmente para os peritos na área, foram desde então complementados por painéis de controlo para a saúde pública ou eleições políticas, concebidos para um público mais geral e divulgados através dos meios de comunicação social ou de sites concebidos para o efeito [Bach et al., 2022]. Nas figuras 2.20, 2.20 e 2.22, apresentamos 3 *dashboards* com fins e públicos-alvo diferentes.

A primeira plataforma é uma ferramenta desenvolvida pela Amazon, chamada QuickSight. O QuickSight é um serviço de *business intelligence* que facilita a construção e publicação de painéis interactivos que fornecem informações sobre os seus dados empresariais. Os *dashboards* podem ser criados ligando-se a várias fontes de dados e depois utilizando ferramentas simples de *drag-and-drop* para construir visualizações [Russo, 2003].

Na figura 2.21, é apresentada uma captura de ecrã da COVID19INDIA, um *dashboard* de análise da progressão do COVID-19 na Índia [Covid19India, 2021]. Este software é explicado em maior detalhe na secção 3.1.

Por último, a figura 2.22. ilustra a página de *analytics* da rede social Instagram. Este *dashboard* fornece informação detalhada sobre o desempenho de uma conta de Instagram. Permite aos utilizadores acompanhar métricas como gostos, comentários e seguidores ao longo do tempo, bem como ver quais as publicações que estão a ter o melhor desempenho.

2.3 Tarefas de Utilizador

Para atingirmos os objetivos definidos por uma dada visualização, precisamos primeiro de definir a maneira como se processa. Ao servir-se de uma visualização, o utilizador pode desempenhar 3 ações principais: *Analisar, Pesquisar, e Consultar* [Munzner, 2014]. Tendo em conta estes objetivos, estabelece-se assim uma simbiose a que chamamos Discurso Analítico: a relação abstrata (i.e não-específica ao domínio) entre as tarefas (ou ações) do utilizador e as técnicas de interação que fornecem as respostas às suas questões. Assim, o Discurso Analítico define a maneira como o humano integra e é integrado na visualização em si. A exploração da visualização depende inteiramente da perceção e conhecimento prévio do utilizador, que coloca questões, formula hipóteses e apoia-se na ferramenta para obter respostas [Brehmer et al., 2014]. No domínio específico dos da-



Figura 2.20: Amazon QuickSight [Bach et al., 2022].

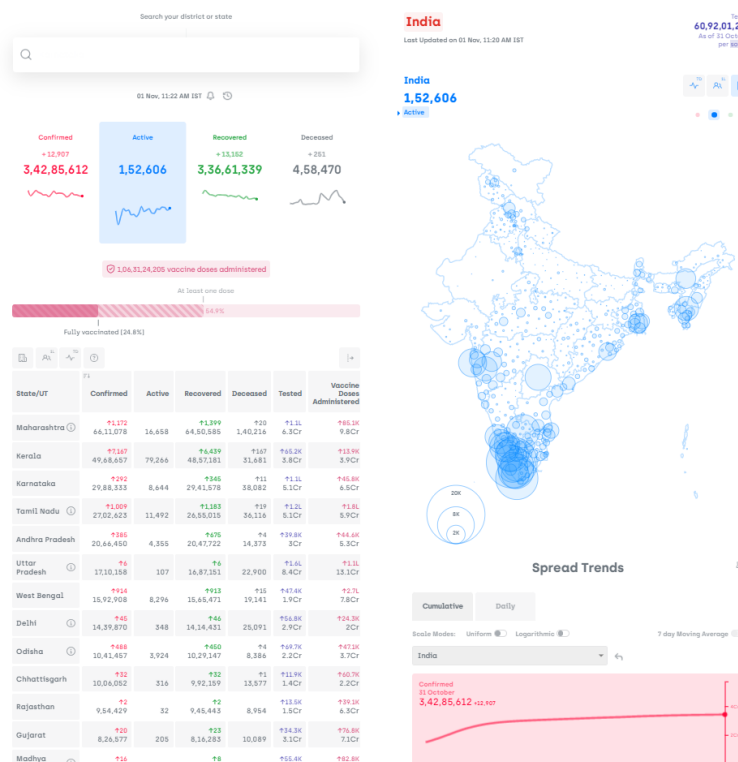


Figura 2.21: COVID19INDIA [Bach et al., 2022].



Figura 2.22: Dashboard do Instagram Analytics [Bach et al., 2022].

dos espaço-temporais, identificaram-se três componentes fundamentais: espaço (onde), tempo (quando), e objecto (o quê) [Peuquet, 2002]. A autora propôs uma topologia para tipos de questões que surgem da combinação destes componentes:

- "Que objecto(s) estão presentes num determinado local e num determinado momento?", i.e. *onde e quando?*
- "Onde está um determinado objecto(s) localizado(s) num determinado momento?", i.e. *quando e o quê?*
- "Quando é que um determinado objecto(s) se encontra(m) num determinado local?", i.e. *onde e o quê?*

Mais recentemente, foi desenvolvido um sistema para a classificação de tarefas, que se baseia nas intenções dos utilizadores e no papel das técnicas de interacção. As tarefas são classificadas em sete categorias [Yi et al., 2007]:

1. **Selecionar** - Marcar um item como relevante para o monitorizar e acompanhar.
2. **Explorar** - Mostrar um (ou vários) sub-conjunto(s) dos dados representados.
3. **Reconfigurar** - Alterar a disposição espacial dos elementos para fornecer uma perspectiva diferente sobre os dados.
4. **Codificar** - Fornecer uma representação visual fundamentalmente diferente, seleccionando outra técnica ou modelo de visualização.
5. **Abstrair** - Mostrar níveis diferentes de detalhe.
6. **Filtrar** - Visualizar certos itens que seguem uma certa restrição condicional.

7. **Interligar** - Mostrar ligações visuais entre itens relacionados, em várias vistas diferentes.

Este sistema de classificação estabelece o conceito de que as tarefas podem, na sua maioria, ser divididas em tarefas de alto e baixo nível, e que estes níveis têm uma ligação directa às técnicas de interacção. Esta relação entre os objectivos e tarefas do utilizador e as técnicas de interacção de baixo nível forma o discurso analítico acima mencionado.

2.4 Técnicas de Interação

As técnicas de interacção desempenham um papel crucial no processo de desenho de visualizações e interfaces do utilizador eficazes e intuitivas. Estas técnicas permitem aos utilizadores manipular e interagir com os dados e com as visualizações, e podem influenciar grandemente a usabilidade e utilidade de um sistema. Nesta secção, damos uma visão geral de um pequeno subconjunto das várias técnicas de interacção que têm vindo a ser desenvolvidas para a visualização de informação e interfaces de utilizador.

2.4.1 Overview plus Detail

O *Overview* destina-se a dar ao utilizador um amplo conhecimento de todo o espaço de informação e cujo objectivo principal é resumir. Um objectivo comum na concepção da visão geral é mostrar todos os itens do conjunto de dados em simultâneo, sem qualquer necessidade de navegação ou *scroll*. As visões gerais ajudam o utilizador a encontrar regiões da visualização onde uma investigação mais detalhada pode ser produtiva. As visões gerais são frequentemente mostradas no início do processo de exploração, para orientar os utilizadores na escolha do local a pesquisar para inspeccionar com mais detalhe. No entanto, a utilização da visão geral não se limita ao reconhecimento inicial; é muito comum os utilizadores intercalarem a utilização de visões gerais e de detalhe, alternando muitas vezes entre elas. O objectivo desta técnica é mostrar informação com diferentes níveis de detalhe em múltiplas visões ligadas [Card, 1999]. Em particular, uma vista mostra informação detalhada sobre a área relevante, enquanto a(s) outra(s) vista(s) mostra(m) uma visão geral do problema [Keim et al., 2008; Munzner, 2014]. No exemplo da figura 2.23., vemos uma captura de ecrã do Visual Studio Code, um editor de texto desenvolvido pela Microsoft. Nesta imagem, podemos ver duas vistas diferentes - o editor de texto propriamente dito, que apresenta o código com maior granularidade e que corresponde ao *Detail*, e à direita, um mini-mapa do código, que nos dá uma visão geral do código escrito, compactando a informação numa área mais pequena - isto é, um *Overview*.

```

29 export const NoteFormMenu = () => {
36   const { isOpen: isOpenEdit, onClose: onCloseEdit, onOpen: onC
37
38   > React.useEffect(() => { ...
41     }, [isEditing, setTempCollapsed]);
42
43   > const canProcessNote = React.useMemo( ...
46     );
47
48   > const isLocked = React.useMemo( ...
58     );
59
60   const formatDate = (date: string) => (date ? " at " + format(
61
62   > const getNoteStatus = () => { ...
89     });
90
91   > const onCancel = async () => { ...
94     });
95
96   > const onConfirmEdit = async () => { ...

```

Figura 2.23: Captura de ecrã do editor de texto VSCode. Neste exemplo, temos uma visão geral do código todo sob a forma de um mini-mapa (do lado direito) e de uma secção particular do lado esquerdo.

2.4.2 Linking and Brushing

O *Linking and Brushing* combina diferentes métodos de visualização para superar as deficiências de técnicas únicas. Os pontos seleccionados (i.e. *brushed*) são realçados em todas as visualizações - que estão ligadas entre si -, tornando possível detectar dependências e correlações. As alterações interactivas efectuadas numa visualização são automaticamente reflectidas nas outras visualizações. Note-se que a ligação de múltiplas visualizações com recurso a esta técnica fornece mais informação do que a consideração das componentes de visualização de forma independente [Keim, 2002].

2.4.3 Focus plus Context

A ideia subjacente a esta técnica é a de permitir ao utilizador ver regiões da visualização de interesse primário com maior detalhe - i.e. *Focus* -, mantendo a visão geral da informação circundante, em menor detalhe sobre a mesma visão - i.e. *Context* [Baudisch et al., 2002]. Implementações desta técnica, tais como *fisheye lens*, permitem aos utilizadores visualizar partes de documentos seleccionados com mais detalhes sem necessidade de uma segunda janela [Polisciuc, 2021].

A figura 2.24 é uma captura de ecrã da barra de tarefas do sistema operativo dos computadores da Apple, o MacOS. Este exemplo ilustra o conceito de *Focus* - dando maior relevância ao elemento que é objeto de foco com o rato -, e *Context* - permitindo que o utilizador continue a ver os restantes componentes da barra de tarefas [Cockburn et al., 2009].

A técnica *fisheye lens*, previamente mencionada, está representada pela figura 2.25.

A técnica de interação retratada faz parte do caso de estudo *Data Lenses*, que integra a iniciativa *LIVE Singapore*, desenvolvida pelo Senseable Lab, do MIT [Kloeckl et al., 2012]. Esta solução procura uma maneira de visualizar dados relativos à circulação dos cidadãos singapurenses com recurso ao autocarro. Esta implementação permite distorcer o espaço geográfico circundado pela lente, de maneira a que se consiga distinguir dados com uma concentração muito elevada, sem perder de vista o contexto em que se inserem. A lente actua como uma lupa, permitindo ao utilizador descobrir camadas de informação que revelam a atividade da rede de autocarros de Singapura com maior detalhe.

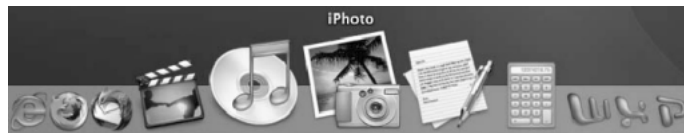


Figura 2.24: Barra de tarefas do MacOS. Neste exemplo, é possível dar maior destaque a um ícone específico, mantendo em vista os restantes ícones que compõem a barra de tarefas [Cockburn et al., 2009].

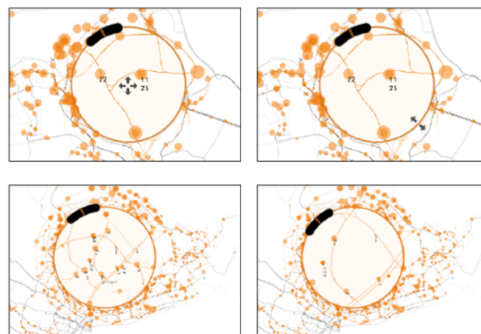


Figura 2.25: Técnica de *fisheye lens* [Kloeckl et al., 2012].

2.4.4 Panning and Zooming

A navegação pode ser decomposta em três componentes. A ação de *zoom* aproxima ou afasta a câmara do plano observado. Aproximar a câmara do plano - i.e. *zoom in* -, mostra menos itens, que aparentam ser maiores. Pela mesma lógica, afastar a câmara do plano - i.e. *zoom out* -, mostra mais itens, que parecem mais pequenos. Quando se faz *panning*, a câmara move-a paralelamente ao plano da imagem, para cima e para baixo ou de um lado para o outro. Esta técnica é uma maneira de combater o espaço de ecrã limitado [Munzner, 2014]. Este é precisamente o tipo de comportamento que vemos em plataformas como a que está representada na figura 2.26, o Google Maps [Google, 2005].



Figura 2.26: Página principal da interface do Google Maps. Esta plataforma permite ao utilizador arrastar com o rato o seu campo de visão para outras localizações geográficas, focando em áreas diferentes. Para além disso, é ainda possível fazer zoom-in ou zoom-out, aumentando ou reduzindo a abrangência do seu campo de visão.

2.5 Tecnologias

A escolha das tecnologias certas para auxiliar e possibilitar o processo de desenvolvimento do projeto requer uma abordagem cuidada e sistemática. Nesta secção apresentamos algumas ferramentas e tecnologias populares que dizem respeito às quatro principais componentes do desenvolvimento da plataforma:

- Ferramentas de pré-processamento e limpeza dos dados: Tem que possibilitar a manipulação fácil e rápida dos dados pré-existent.
- Plataformas *Backend-as-a-Service*: No fundo, é apenas uma base de dados. Pode estar hospedada localmente, ou na *cloud*. Possibilita a pesquisa sobre os dados e seu armazenamento de forma fácil.
- *Frameworks* de *UI*: Ferramentas que tornam possível a construção de interfaces de utilizador complexas, com recurso ao paradigma da modularidade previamente discutido.
- Ferramentas de visualização para *web*: Bibliotecas que permitam a renderização de visualizações simples num *browser*.

2.5.1 Pré-processamento de dados

As ferramentas de pré-processamento de dados são componentes essenciais do conjunto de ferramentas da ciência dos dados e de machine learning. Estas tecnologias proporcionam aos programadores e analistas de dados funcionalidades poderosas para manipular, limpar e preparar dados antes de serem utilizados para análise ou treino de modelos. Estas ferramentas desempenham um papel crucial na transformação de dados em bruto num formato bem estruturado e organizado que pode ser facilmente analisado e utilizado para várias tarefas baseadas em dados.

NumPy

NumPy, abreviatura de "Numerical Python", é uma biblioteca open-source fundamental para a computação numérica em Python. O NumPy é a base de muitas outras bibliotecas científicas e relacionadas com dados no ecossistema Python - ver secção 2.5.1 - *Pandas* abaixo. Lançado em 2006, esta biblioteca tornou-se desde então uma ferramenta padrão para lidar com vetores, matrizes e várias funções matemáticas em Python, oferecendo um paradigma de programação orientado para vetores que simplifica cálculos numéricos complexos.

A característica principal do NumPy é o seu objeto matriz multidimensional, conhecido como "NumPy array" ou ndarray. O ndarray fornece uma estrutura de dados versátil para representar dados homogêneos e multidimensionais, suportando vetores de diferentes formas e tamanhos. As operações de vetores do NumPy são significativamente mais rápidas e mais eficientes em termos de memória do que as listas tradicionais do Python, tornando-o uma escolha ideal para cálculos numéricos. A biblioteca oferece uma vasta gama de funções matemáticas para operações de elementos, álgebra linear, estatística e muito mais, que podem ser aplicadas a arrays inteiros com facilidade.

Pandas

O Pandas é uma biblioteca open-source escrita em Python, concebida para a manipulação e análise de dados. Construído sobre a biblioteca NumPy, o Pandas fornece estruturas de dados poderosas e eficientes, tornando-o uma ferramenta indispensável para trabalhar com dados estruturados. O nome "Pandas" deriva de "panel data", um termo utilizado em econometria para se referir a conjuntos de dados que contêm observações ao longo de vários períodos de tempo para várias entidades. Desde o seu lançamento em 2008, o Pandas tornou-se uma solução de eleição para tarefas de manipulação de dados devido à sua simplicidade, versatilidade e extensa funcionalidade.

O funcionamento do Pandas assenta em duas estruturas de dados principais: Series e DataFrame. Uma Series é um vetor unidimensional que pode conter vários tipos de dados, como números inteiros, floats, strings e até mesmo objetos personalizados. Por outro lado, um DataFrame é uma estrutura de dados bidimensional semelhante a uma tabela, onde os dados são organizados em linhas e colunas, e cada coluna pode conter diferentes tipos de dados. O DataFrame é particularmente poderoso para lidar com dados tabulares e oferece uma infinidade de operações para filtragem, limpeza, fusão, agrupamento e agregação de dados. Com a sua sintaxe intuitiva e API abrangente, o Pandas permite aos utilizadores manipular dados de forma eficiente, efetuar cálculos estatísticos e gerar conhecimentos significativos a partir de grandes conjuntos de dados.

Uma das características mais proeminentes do Pandas é a sua excepcional documentação. A documentação oficial do Pandas é abrangente, bem organizada e actualizada regularmente, fornecendo explicações claras sobre funções, métodos e conceitos da biblioteca. Inclui vários exemplos e tutoriais, o que a torna um recurso valioso tanto para iniciantes como para cientistas de dados experientes.

Além disso, a comunidade ativa do Pandas contribui para o seu sucesso, submetendo regularmente correções de erros, pedidos de funcionalidades e extensões inovadoras. Este esforço de colaboração garante que a biblioteca se mantém atualizada, fiável e melhora continuamente a cada nova versão. A forte documentação e o envolvimento da comunidade fazem do Pandas a escolha preferida para tarefas de manipulação de dados em Python, solidificando a sua posição como uma biblioteca fundamental no ecossistema da ciência de dados

Danfo.js

O Danfo.js é uma *package* do *NPM* que providencia estruturas de dados rápidas, flexíveis e expressivas, concebidas para tornar o trabalho com dados relacionais fácil e intuitivo. É fortemente inspirado pela biblioteca Pandas e fornece uma API semelhante. Isto significa que os utilizadores familiarizados com o Pandas podem facilmente utilizar o danfo.js.

No entanto, a maior diferença entre o danfo.js e o Pandas é mesmo a linguagem na qual estão escritas. O danfo.js é escrito em TypeScript e JavaScript e foi criado com o objetivo de possibilitar manipulação de dados na *web*. Esta tecnologia é relativamente recente, tendo sido lançada em 2022, e como tal, não tem uma comunidade tão grande como tecnologias como o Pandas ou o NumPy, que são líderes do setor. Para além disso, o conjunto de funcionalidades que o danfo.js oferece é extremamente reduzido quando comparado a um gigante como o Pandas.

No entanto, o danfo.js encontra a sua utilidade em pequenas tarefas de processamento de dados que sejam necessárias - p.e. pré-processar um ficheiro .csv após upload para o servidor -, e é particularmente útil quando a *tech stack*¹ de um determinado projeto é composta por ferramentas escritas totalmente em JavaScript ou TypeScript e optar por pré-processamento em Python com Pandas é de difícil integração.

2.5.2 Backend-as-a-Service

Backend-as-a-Service (ou *BaaS*), é um modelo de *cloud computing* que fornece aos programadores um conjunto abrangente de serviços e funcionalidades de backend através de APIs. As plataformas BaaS abstraem as complexidades da construção e gestão da infraestrutura do lado do servidor, permitindo que os programadores se concentrem no desenvolvimento do front-end e na experiência do utilizador. Estes tipos de plataformas revolucionaram a forma como os programadores criam e implementam aplicações Web e móveis, fornecendo uma solução conveniente e eficiente para gerir a infraestrutura e os serviços de backend. Democratizou o desenvolvimento de aplicações, permitindo aos programadores

¹*Tech Stack* - Combinação de tecnologias de software, linguagens de programação, estruturas e ferramentas que são utilizadas para criar e executar uma determinada aplicação ou sistema de software. Abrange todo o conjunto de tecnologias e componentes necessários para desenvolver, implementar e manter a aplicação ao longo do seu ciclo de vida.

de todos os níveis de competências criar aplicações sofisticadas e ricas em funcionalidades sem necessidade de conhecimentos profundos de backend.

Firestore

O Firestore é uma plataforma criada pela Google para a criação de aplicações *web* e móveis. O Firestore oferece um conjunto de serviços e ferramentas integrados que cobrem os aspectos de frontend e backend do desenvolvimento de software para *web*, tornando-o uma solução completa de backend-as-a-service.

No frontend, é possível utilizar frameworks de *UI* como React, Angular ou Vue.js para criar a interface de utilizador. O Firestore também fornece um SDK para JavaScript que permite a integração perfeita com os serviços de backend da plataforma.

Para o backend, o Firestore oferece vários serviços que gerem a autenticação, a base de dados em tempo real, o *cloud storage*, as *cloud functions* e muito mais. O serviço de autenticação permite aos programadores implementar autenticação e autorização de forma segura. A base de dados em tempo real é uma base de dados NoSQL - i.e. *document-based* -, que sincroniza dados em tempo real, permitindo aplicações reactivas. O serviço de *cloud storage* fornece armazenamento seguro e escalável, enquanto que as *cloud functions* permitem que os utilizadores executem *serverless functions*.

Além disso, o Firestore oferece ferramentas de análise, monitorização de desempenho e relatórios de falhas que fornecem informações valiosas sobre a utilização e o desempenho das aplicações.

Uma das principais vantagens da utilização do Firestore é a sua facilidade de utilização e as capacidades de desenvolvimento rápido. Ao abstrair grande parte da complexidade do back-end, os programadores podem concentrar-se na criação de funcionalidades e em proporcionar uma excelente experiência de utilizador sem terem de gerir a infraestrutura. No entanto, esta plataforma não possibilita o *self-hosting*². Adicionalmente, o paradigma de base de dados *document-based* utilizado pelo NoSQL pode não ser útil na grande maioria das situações em que se trabalha com dados tabulares.

Supabase

O Supabase é uma plataforma *open-source* de Backend-as-a-Service (BaaS) que tem como objetivo simplificar e acelerar o processo de construção de aplicações com uma infraestrutura de backend. Fornece aos programadores um conjunto de ferramentas e serviços que lhes permitem criar, gerir e escalar aplicações sem a necessidade de desenvolvimento tradicional de backend. O Supabase é construído com base em tecnologias populares *open-source*, incluindo PostgreSQL

²"Self-hosted" refere-se à prática de alojar - i.e. *hosting* -, e gerir uma aplicação, serviço ou plataforma de software em infra-estruturas ou servidores pertencentes e controlados pelo próprio utilizador ou organização.

para a camada de base de dados e PostgREST para a camada de API RESTful. Assim, esta tecnologia emprega um paradigma de bases de dados relacional, diferindo de outras como o Firebase, que é document-based - i.e. NoSQL. Também se integra com outros serviços como Autenticação e File Storage para oferecer uma solução abrangente para o desenvolvimento de aplicações web.

As principais características do Supabase rodam à volta da integração perfeita com frameworks de front-end populares, através de SDKs³ para linguagens populares. A camada RESTful API do Supabase é gerada automaticamente a partir da base de dados PostgreSQL subjacente, permitindo que os programadores interajam com a base de dados usando endpoints RESTful familiares. Além disso, o Supabase oferece mecanismos de autenticação e autorização integrados, facilitando a segurança e a gestão do acesso dos utilizadores aos dados. O painel de administrador da plataforma é também digno de ser referido, pois possibilita uma visão transparente para dentro das bases de dados, auxiliada por uma interface limpa e fácil de utilizar. A plataforma é ainda passível de ser *self-hosted*, permitindo ao administrador controlo total dos recursos e impedindo o *vendor lock-in*⁴.

Um dos principais pontos fortes do Supabase reside no apoio da comunidade que rodeia esta tecnologia. Até à data, o Supabase suporta integração, através de SDKs, com seis linguagens distintas - JavaScript/TypeScript, Flutter, Python, C#, Swift e Kotlin. Destas seis, apenas duas foram desenvolvidas pela empresa em si (JavaScript/TypeScript e Flutter), tendo as restantes sido desenvolvidas pela comunidade. Para além disso, a equipa por detrás do Supabase envolve-se ativamente com a comunidade através de vários canais (i.e. GitHub e StackOverflow), respondendo às questões e feedback dos utilizadores. A documentação é algo incompleta, nomeadamente a nível da interação com o painel de administrador. No entanto, a plataforma apresenta um conjunto extenso de tutoriais e exemplos dos *use-cases* mais comuns em *web development*. Assim, a combinação de uma documentação decente e envolvimento da comunidade garante que o Supabase se mantém fácil de usar e acompanha a evolução das necessidades dos programadores.

2.5.3 Frameworks de UI

As frameworks de UI modernas são uma classe de ferramentas de software que facilitam a criação de interfaces de utilizador visualmente apelativas e fáceis de utilizar para aplicações Web e móveis. Estas tecnologias tornaram-se essenciais para os programadores que procuram criar aplicações modernas, reactivas e ricas em funcionalidades com fluxos de trabalho de desenvolvimento eficientes.

³SDK ou *Software Development Kit* é um conjunto de ferramentas, bibliotecas, documentação e exemplos de código que os programadores utilizam para criar aplicações ou software para uma plataforma, um sistema operativo ou uma estrutura específicos. Os SDKs simplificam o processo de desenvolvimento, fornecendo funcionalidades e componentes pré-construídos que podem ser integrados num dado projeto, poupando tempo e esforço na escrita de código de raiz.

⁴"Vendor lock-in" refere-se a uma situação em que um cliente se torna fortemente dependente dos produtos, serviços ou tecnologias de um determinado fornecedor, tornando difícil ou dispendiosa a mudança para um fornecedor ou solução alternativa.

Ao fornecerem um conjunto de componentes de UI pré-concebidos, estilos e elementos interactivos, as frameworks de UI simplificam o processo de design e desenvolvimento, permitindo aos programadores concentrarem-se na funcionalidade e na experiência do utilizador.

React

O React.js, também conhecido apenas por React, é uma *framework* open-source JavaScript ou TypeScript desenvolvida pelo Facebook. É amplamente utilizada para criar interfaces de utilizador para aplicações Web e móveis. O React defende uma arquitetura baseada em componentes, em que os elementos da UI são divididos em componentes reutilizáveis e independentes. As principais características do React revolucionaram a forma como os programadores constroem UIs. Esta secção apresenta as principais funcionalidades do React e o seu funcionamento interno.

JSX O JSX - ou *JavaScript XML* -, é uma extensão da sintaxe utilizada no React para descrever a estrutura e a composição das interfaces de utilizador. O JSX permite escrever código semelhante a HTML dentro de código JavaScript, facilitando a definição da estrutura dos componentes e sua lógica de renderização. Por outras palavras, em vez de utilizar JavaScript para criar e manipular elementos DOM, o JSX permite escrever *tags* semelhantes a HTML que representam os componentes que se pretende renderizar. O código JSX é então transformado em código JavaScript normal por transpilers antes de ser executado no navegador. A figura 2.27 contém dois excertos de código React válido, que, quando executados, têm exatamente o mesmo resultado - um elemento do tipo *h1* (um cabeçalho HTML), com a classe *greeting*, e conteúdo *Hello world!*. O elemento *componentA* está escrito utilizando código JavaScript puro, e o *componentB* com recurso a JSX. É impossível não reparar nas claras diferenças entre os dois exemplos - um mais prolixo e difícil de ler, outro mais simples e legível -, tornando fácil perceber o porquê da sua adoção e uso generalizado.

```
const componentA = React.createElement(  
  'h1',  
  { className: 'greeting' },  
  'Hello world!'  
)  
  
const componentB = (  
  <h1 className='greeting'>  
    Hello world!  
  </h1>  
)
```

Figura 2.27: Exemplo do mesmo elemento React criado com recurso a JavaScript puro (em cima), e com JSX (em baixo)

DOM Virtual A DOM, ou *Document Object Model*, é uma norma definida pelo W3C para aceder ao código HTML ou XML em linguagens de *scripting*. A DOM ajuda a manipular dados num modelo orientado a objectos porque os elementos da DOM têm uma estrutura definida em objectos, métodos ou propriedades que podem ser facilmente acedidos. São tratados como nós e representados como uma árvore.

Uma das suas principais inovações do React é a DOM (Document Object Model) virtual. A DOM virtual é uma representação rápida e em memória da DOM real, e é uma abstracção que nos permite tratar o JavaScript e a DOM como se fossem reativos. Efetivamente, sempre que o estado dos dados muda, a DOM virtual e o React renderizam novamente a interface de utilizador, criando uma representação virtual da DOM. De seguida, o React calcula a diferença entre duas representações virtuais da DOM virtual: a representação da DOM virtual anterior, que foi computada antes dos dados serem alterados, e a representação da DOM virtual atual, que foi computada depois dos dados serem alterados. Essa diferença entre as duas representações virtuais da DOM é o que realmente necessita de ser alterado na DOM real.

Finalmente, o React atualiza apenas o que precisa de ser atualizado na DOM real 2.28. O processo de encontrar diferenças entre as duas representações da DOM virtual e re-renderizar apenas as secções da UI que mudaram na DOM real é rápido [Fedosejev, 2015].

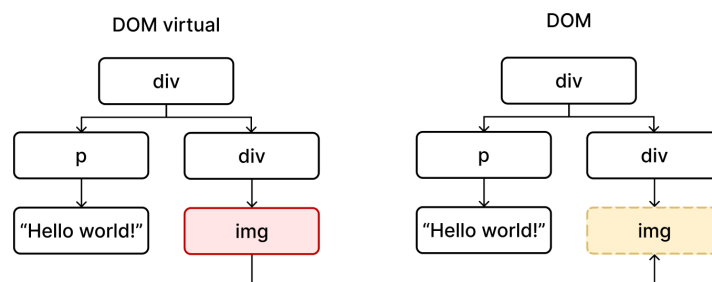


Figura 2.28: Atualização da DOM através da DOM virtual. O elemento *img* mudou, e portanto é atualizada na DOM real. Apenas este componente é atualizado, com o resto dos componentes da página a manterem-se inalterados [Fedosejev, 2015].

Componentes A filosofia de desenvolvimento do React assenta na ideia de que qualquer interface de utilizador é descomponível nas partes mais pequenas que a constituem, chamadas componentes. A separação de uma interface em componentes ajuda a gerir melhor o código, uma vez que é mais fácil de manter. A figura 2.29 mostra a decomposição de uma interface de utilizador básica - neste caso, uma aplicação de gestão de *stock* -, nos componentes que a integram. O exemplo, retirado da documentação oficial do React⁵, mostra-nos como pensar numa interface de utilizador como a soma de partes.

⁵Documentação oficial do React - *Thinking in React*

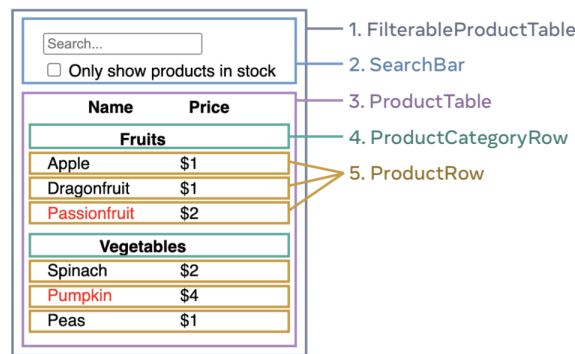


Figura 2.29: Exemplo da decomposição da interface de utilizador de uma aplicação de gestão de stocks em componentes [Fedosejev, 2015].

Estado No React, o estado refere-se aos dados que um componente pode manter e gerir internamente. Representa as informações dinâmicas que podem mudar com o tempo e influenciar a renderização e o comportamento do componente. O estado permite que os componentes React respondam às interações do usuário, alterações de dados e outros eventos, atualizando e renderizando novamente partes da interface do usuário. O estado é um conceito essencial na lista de conceitos do React, pois sem ele, os componentes seriam renderizados apenas uma vez e permaneceriam estáticos. Ao introduzir o estado, os componentes podem mudar o que renderizam com base em alterações nos dados que possuem.

O React surgiu como uma das estruturas de UI mais populares e amplamente adoptadas no mercado. A sua simplicidade, escalabilidade e amplo apoio da comunidade contribuíram para a sua utilização generalizada. O ecossistema do React é enriquecido com inúmeras bibliotecas, ferramentas e extensões, tornando-o altamente versátil e adaptável a vários requisitos de projeto. Concorre favoravelmente com outras tecnologias como Angular e Vue.js, oferecendo aos programadores uma solução flexível e poderosa para a criação de aplicações Web e móveis modernas.

Vue

O Vue (pronuncia-se "view") é uma framework JavaScript progressiva para a criação de interfaces de utilizador. É um projeto independente e orientado para a comunidade. Foi criado por Evan You em 2014 como um projeto pessoal. Hoje, o Vue é mantido ativamente por uma equipa de membros a tempo inteiro e voluntários de todo o mundo [You, 2023].

O objetivo do Vue.js é fornecer os benefícios da ligação reactiva de dados e componentes visuais com uma API tão simples quanto possível. O Vue.js em si não é uma framework completa - está focado apenas na camada de visualização, i.e. o V em MVC. Por conseguinte, é muito fácil de utilizar e integrar com outras bibliotecas ou projectos existentes. Por outro lado, quando usado em combinação com ferramentas adequadas e bibliotecas de suporte, o Vue.js também é perfeitamente capaz de produzir uma Single Page Application sofisticada [Kyriakidis

et al., 2016].

Vue CLI O Vue CLI é a ferramenta padrão para desenvolvimento em Vue e vem com suporte para Babel, TypeScript, pré-processadores CSS e ESLint, bem como testes unitários e testes end-to-end. O Vue CLI também tem uma GUI opcional que pode ser utilizada para criar e gerir projectos. Apesar do seu uso não ser estritamente necessário, esta ferramenta melhora bastante a experiência de utilizador⁶

Vue Single-File Components Os Single-File Components (SFCs) do Vue são ficheiros com extensão `.vue` que consistem num *template* com três blocos: Um para descrever a UI com HTML, um bloco de *scripting* para adicionar lógica com JavaScript e um bloco de estilos para estilizar o componente com CSS (figura 2.30). Os SFCs também têm suporte para *scoped CSS* e módulos *webpack* que permitem o uso das instruções `import` e `@import` dentro de blocos de lógica e estilo, respetivamente. Os comentários podem ser usados dentro dos três blocos na respectiva sintaxe de comentários da linguagem usada, enquanto os comentários de nível superior seguem as convenções dos comentários HTML.

```

<template>
  <p class="example-class">{{ exampleProperty }} World!</p>
</template>

<script>
export default {
  data() {
    return exampleProperty: 'hello world!'
  }
};
</script>

<style scoped>
.example-class {
  example-property: example-value;
}
</style>

```

Figura 2.30: Exemplo básico da estrutura de um SFC, composto pelo bloco de HTML (`<template />`), pelo bloco de lógica (`<script />`), e pelo bloco de *styling* (`<style />`)

O bloco HTML permite que a DOM renderizado seja vinculado declarativamente aos dados subjacentes do bloco de lógica, o que, por sua vez, permite que o Vue determine de forma inteligente o número mínimo de componentes a serem renderizados novamente e as manipulações da DOM a serem aplicadas para acomodar uma mudança de estado. A interpolação de texto com a *mustache syntax*, denotada por `{{ }}`, é a forma mais básica da referida *data binding* e pode ser usada

⁶A experiência de utilizador (ou DX, do inglês *Developer Experience*) refere-se à experiência geral que os programadores têm quando trabalham com uma linguagem de programação, framework, biblioteca, ferramenta ou plataforma. A DX engloba vários aspectos da interação de um programador com a tecnologia, incluindo a facilidade de utilização, a eficiência, a produtividade, a satisfação e a qualidade geral do processo de desenvolvimento. Essencialmente, a DX centra-se na criação de um ambiente positivo e eficiente para os programadores, permitindo-lhes escrever código, corrigir, testar e implementar aplicações com facilidade e eficácia. Uma boa DX pode levar a ciclos de desenvolvimento mais rápidos, curvas de aprendizagem reduzidas, menos erros e maior colaboração entre os membros da equipa.

para definir dinamicamente um texto dentro do bloco de modelo para corresponder à propriedade de dados correspondente do bloco de script. Um SFC só pode conter um único bloco HTML.

O bloco de lógica (baseado em JavaScript) é utilizado para incluir um objeto de opções, que contém informações sobre os dados e a lógica (por exemplo, propriedades e métodos de dados) do componente. A propriedade `data` do objeto de opções é uma função que devolve um objeto, que é depois chamado quando o estado muda e referenciado quando a vista está a ser renderizada novamente. Um SFC só pode conter um único bloco de lógica de cada vez. O bloco de estilo baseado em CSS é usado para estilizar elementos dentro do bloco de modelo e pode ser encapsulado na scope do componente com `scoped CSS`. Um SFC pode conter vários blocos de estilo ao mesmo tempo com declarações CSS locais e globais.

2.5.4 Visualização

As bibliotecas de visualização permitem que os programadores e os designers criem visualizações orientadas para os dados que facilitam a sua exploração. Nesta secção, analisamos o panorama das ferramentas e bibliotecas de visualização especificamente concebidas para aplicações *web*. Apresentamos três soluções de visualização com graus de abstração, integrações e focos diferentes, numa tentativa de ilustrar um pouco da paisagem das ferramentas de visualização disponíveis no *web development*.

D3.js

O D3.js, ou Data-Driven Documents, é uma biblioteca JavaScript para criar visualizações interactivas e baseadas em dados em navegadores Web. Desenvolvido por Mike Bostock, o D3.js permite aos programadores associar dados à DOM⁷; por conseguinte, é possível efetuar manipulações de dados, bem como transformações baseadas em dados, no documento.

O D3 cria visualizações através de Scalable Vector Graphics (SVG), HTML e Cascaded Style Sheets (CSS) e lida com diferentes formatos de dados, incluindo CSV, JSON e GeoJSON. É uma ferramenta gratuita e open-source, mas requer conhecimentos básicos de HTML, JavaScript, CSS e SVG para criar gráficos. A escrita de funções de baixo nível a partir do zero é necessária mesmo para criar visualizações simples, o que é relativamente difícil e prolonga consideravelmente o tempo de desenvolvimento, uma vez que o D3 não dispõe de uma biblioteca pré-construída de tabelas ou gráficos à escolha. No entanto, estão disponíveis

⁷A Document Object Model, ou simplesmente DOM, é uma interface de programação e uma representação da estrutura e do conteúdo de uma página Web de forma hierárquica. Em termos mais simples, a DOM pode ser considerada como uma estrutura em forma de árvore em que cada nó representa um elemento numa dada página Web. Estes elementos podem incluir *tags* de HTML, conteúdo de texto, imagens, formulários, entre outros. A DOM permite que os programadores acedam e modifiquem estes elementos de forma dinâmica, possibilitando a criação de aplicações Web interactivas e dinâmicas.

funções reutilizáveis e amostras de gráficos comuns⁸, suportados pela equipa do D3 e pela comunidade.

No cerne do D3.js está a seleção - um grupo de elementos seleccionados retirados do documento. Os operadores interagem com as seleções, influenciando o conteúdo. As junções de dados associam dados a elementos HTML, habilitando operadores funcionais que dependem de dados e formando subselecções para criação e remoção de elementos alinhados com os dados. Por defeito, os operadores têm um efeito instantâneo, mas as transições animadas interpolam graciosamente atributos e estilos ao longo do tempo. Os *event handlers*⁹, operadores distintos, respondem ao input do utilizador e facilitam a interação. Além disso, o D3 fornece uma gama de módulos utilitários, incluindo layouts e escalas, simplificando os procedimentos típicos de visualização [Bostock et al., 2011].

Select O D3 adopta a API de selectores do W3C para identificar elementos HTML para seleção. Esta mini-linguagem consiste em predicados que filtram elementos por etiqueta (`tag`), classe (`.class`), identificador único (`#id`), atributo (`[name=value]`), contenção (`parent child`), adjacência (`before ~ after`) e várias outras facetas. Os predicados podem ser intersectados (`.a.b`) ou unidos (`.a, .b`), resultando num método de seleção rico mas conciso [Bostock et al., 2011].

O D3.js fornece o namespace global, `d3`, que oferece os métodos `select` e `selectAll` para reunir seleções. Estes métodos acomodam a mini-linguagem do seletor, com `select` a visar o primeiro elemento correspondente e `selectAll` a englobar todas as correspondências na ordem do documento. Estes métodos também podem receber referências directas a nós, facilitando o acesso a nós de fontes externas.

Os elementos seleccionados podem ser submetidos a vários operadores, tirando partido da API DOM do W3C para manipulação de atributos, estilos, propriedades, HTML e texto. Os valores dos operadores são constantes ou funções avaliadas para cada elemento. Embora os operadores incorporados sirvam para a maioria dos objectivos, o operador `each` invoca chamadas de retorno JavaScript personalizadas. Dada a natureza matricial das seleções, os elementos também podem ser acedidos directamente através dos seus índices (p.e. `element[0]`).

O D3 suporta o encadeamento de métodos, simplificando a aplicação de vários operadores, com o valor de retorno do operador a servir de seleção. Por exemplo, a figura 2.31 representa a criação de uma escala linear utilizando este padrão de encadeamento de comandos. Adicionalmente, operadores como `append` e `insert` adicionam novos elementos para cada um na seleção atual, facilitando convenientemente as estruturas aninhadas. Por outro lado, o operador `remove` elimina os elementos seleccionados.

Enquanto os métodos de seleção de nível superior consultam todo o documento,

⁸D3 Graph Gallery - <https://d3-graph-gallery.com/>

⁹Os eventos são sinais que são despoletados dentro da janela do browser que notificam as alterações no ambiente do browser ou do sistema operativo. Os programadores podem criar código que manipula eventos que será executado quando um evento for acionado, permitindo que as páginas Web respondam adequadamente às alterações [MDN, 2023].

```
const y = scaleLinear()  
  .range([height, 0])  
  .domain([0, max(bins, (d) => d.length) as number]);
```

Figura 2.31: Excerto de código que exemplifica o encadeamento de métodos possibilitado pelo D3. Neste caso, é criada uma escala linear, mapeando o intervalo de pixels `[height, 0]` ao domínio de dados descrito no comando `domain`.

os operadores `select` e `selectAll` de uma seleção restringem as consultas aos descendentes de cada elemento selecionado; chamamos a isto subselecção. Por exemplo, `d3.selectAll("p").select("b")` devolve os primeiros elementos a negrito ("b") em cada elemento de parágrafo ("p"). A subselecção através de `selectAll` agrupa elementos por antepassado. Assim, o excerto `d3.selectAll("p").selectAll("b")` agrupa por parágrafo, enquanto que a instrução `d3.selectAll("p b")` devolve uma seleção plana. A subselecção através do método `select` é semelhante, mas preserva os grupos e propaga os dados.

Data O operador `data` no D3 estabelece uma ligação entre os dados de entrada e os nós selecionados. O D3 adopta uma abordagem versátil no processamento de dados, acomodando *arrays* de vários valores como números, *strings* e *objectos*. Assim que os dados são ligados a elementos HTML, são passados como argumento (por convenção, 'd') para operadores funcionais, juntamente com o índice numérico ('i'). Esta configuração alinha-se com os métodos de *array* incorporados no JavaScript. Esta escolha de design simplifica a estrutura de seleção do D3, limitando o agrupamento a um nível e evitando argumentos variáveis. Por defeito, os dados alinham-se com os elementos por índice: o primeiro elemento emparelha-se com o primeiro ponto de referência, e assim sucessivamente. A natureza persistente dos dados permite que estes permaneçam nos nós após nova seleção, evitando a necessidade de utilização recorrente do operador `data`. Isto simplifica as transformações subsequentes e permite a comparação direta entre dados novos e antigos. Além disso, os dados podem ser utilizados para tarefas como a reordenação, com o comando `sort` ou a filtragem de elementos, com o operador `filter` [Bostock et al., 2011].

Mapbox GL JS

A Mapbox GL JS é uma biblioteca JavaScript *open-source* que oferece ferramentas para a criação de mapas interactivos e personalizáveis na Web. O seu principal objetivo é permitir que os programadores criem aplicações dinâmicas e visualmente apelativas baseadas em mapas que possam ser perfeitamente integradas em aplicações *web*. O Mapbox GL JS é construído com base no WebGL, uma tecnologia que permite a renderização de gráficos interativos num *browser*, acelerados por hardware.

O ponto central do Mapbox GL JS é a sua capacidade de construir mapas que respondem ativamente às interações do utilizador, permitindo o *zoom*, *panning* e animações. Esta interatividade é conseguida através de uma série de gestos,

controles de rato e atalhos de teclado, o que torna a navegação intuitiva.

Um dos atributos mais distintivos do Mapbox GL JS é a sua flexibilidade ao permitir a criação de estilos de mapa personalizados. Através da integração com o Mapbox Studio, os programadores podem conceber meticulosamente uma estética de mapa que se alinhe com a marca e a identidade visual das suas aplicações. Esta personalização estende-se a esquemas de cores, rotulagem e coerência temática geral, reforçando uma experiência de utilizador consistente.

A biblioteca alarga ainda mais as suas capacidades para abranger diversas possibilidades de visualização de dados. Os programadores podem integrar sem problemas várias camadas de dados, incluindo marcadores, símbolos, linhas, polígonos e coropléticos. Esta funcionalidade permite a visualização de dados espaciais, promovendo uma representação visualmente rica e informativa de conhecimentos geográficos. A sua capacidade de estilização orientada por dados é uma faceta fundamental, permitindo aos programadores estilizar de forma adaptável as características do mapa com base em atributos ou propriedades inerentes aos dados subjacentes. Isto revela-se particularmente valioso na criação de mapas temáticos que representam visualmente narrativas baseadas em dados.

O Mapbox GL JS obtém a sua eficiência a partir de mosaicos vectoriais, que processam eficientemente os dados do mapa. Os mosaicos vectoriais oferecem representações compactas das características do mapa, facilitando uma representação suave em vários níveis de zoom. Além disso, a biblioteca facilita a integração de serviços de geolocalização e de funcionalidades de navegação, respondendo eficazmente a aplicações que necessitam de localização em tempo real, planeamento de rotas e direcções.

É de salientar a integração perfeita da biblioteca com uma miríade de outras tecnologias e estruturas Web, facilitando a compatibilidade com as pilhas de desenvolvimento preferidas. Esta versatilidade encoraja os programadores a explorar a criação dinâmica de mapas nos seus ambientes tecnológicos existentes.

Leaflet

Leaflet, uma biblioteca JavaScript de código aberto, surge como uma ferramenta formidável no domínio da visualização de mapas baseados na Web, oferecendo um conjunto distinto de características que a diferenciam das suas congéneres. Em primeiro lugar, o Leaflet serve como uma solução leve e versátil para a apresentação de mapas interactivos em aplicações Web. O seu principal objetivo é fornecer aos programadores uma plataforma fácil de utilizar e adaptável para a criação de mapas que se integram perfeitamente em diversos ambientes Web.

A principal distinção do Leaflet reside na sua natureza leve, que o torna adequado para projectos em que a eficiência de recursos é fundamental. Este atributo resulta do design minimalista do Leaflet, que evita funcionalidades desnecessárias, tornando-o particularmente favorável para aplicações que exigem tempos de carregamento rápidos. Esta característica diferencia o Leaflet do Mapbox GL JS, que emprega WebGL para renderização acelerada e pode implicar uma imple-

mentação mais intensiva de recursos.

Ao contrário do Mapbox GL JS, o Leaflet não se baseia em mosaicos vectoriais para a apresentação de dados de mapas. Em vez disso, utiliza mosaicos raster tradicionais, que consistem em imagens pré-renderizadas que representam características do mapa. Embora os mosaicos vectoriais ofereçam maior flexibilidade e eficiência, os mosaicos raster mantêm a compatibilidade com uma gama mais vasta de fontes de dados cartográficos. Isto torna o Leaflet uma opção atractiva para cenários em que os dados ou serviços de cartografia antigos têm de ser integrados sem problemas.

A extensibilidade do Leaflet é outra característica distintiva. Através de um vasto conjunto de plugins e extensões, os programadores podem melhorar a sua funcionalidade principal para satisfazer os requisitos específicos do projeto. Esta adaptabilidade permite a integração de características adicionais, tais como ferramentas de análise geoespacial, camadas de mapas especializadas e interações personalizadas, adaptadas às necessidades específicas da aplicação.

Uma distinção notável entre o Leaflet e o Mapbox GL JS é a dependência do primeiro na DOM para renderização, em oposição à utilização do WebGL pelo segundo. Esta escolha traduz-se em diferentes características de desempenho e pode afetar as capacidades de renderização de ambas as bibliotecas. Enquanto o WebGL capacita o Mapbox GL JS com gráficos acelerados por hardware, a dependência do Leaflet na DOM contribui para a sua compatibilidade com uma gama mais ampla de navegadores e dispositivos.

Orientado para a comunidade e com uma forte ênfase na simplicidade, o Leaflet criou um nicho como uma biblioteca versátil e acessível para a visualização de mapas. A sua flexibilidade, combinada com um extenso ecossistema de extensões, tornou-a uma escolha favorita para os programadores que procuram criar mapas interactivos que se integram perfeitamente num conjunto diversificado de projectos Web.

Capítulo 3

Trabalho Relacionado

Este capítulo tem como principal objetivo analisar e dissecar a fundo algumas soluções existentes no campo da visualização geográfica. É nesta secção que as descrevemos e os problemas que resolvem, salientamos os pontos fracos e fortes, dando também um olhar preliminar para o que se pode aproveitar e/ou melhorar à luz dos conceitos já descritos na secção anterior.

3.1 COVID19INDIA

O COVID-19 foi (e é) um campo fértil de casos de estudo para plataformas de visualização ao serviço da população geral. O software que falamos nesta secção é o arquétipo de soluções *open-source* para benefício do bem-comum e divulgação de informação que nos concerne a todos. Devido à falta de soluções viáveis por parte do governo indiano, a COVID19INDIA¹ surgiu durante o período mais agressivo da pandemia na Índia. Foi desenvolvida por um grupo de voluntários que curaram e verificaram os dados oriundos de várias fontes - desde boletins estatais a inquéritos médicos.

3.1.1 Objetivos

A plataforma serve exclusivamente para acompanhar a evolução da pandemia ao longo do tempo. Este objetivo advém da necessidade de “documentar os dados para que as pessoas com conhecimento possam utilizar estes dados para tomar decisões informada” [Covid19India, 2021].

3.1.2 A Solução

A página principal da plataforma (figura 3.1) tem cinco elementos principais que saltam à vista: uma barra de pesquisa, os gráficos *sparkline*, o mapa do país, os

¹<https://www.covid19india.org/>

Capítulo 3

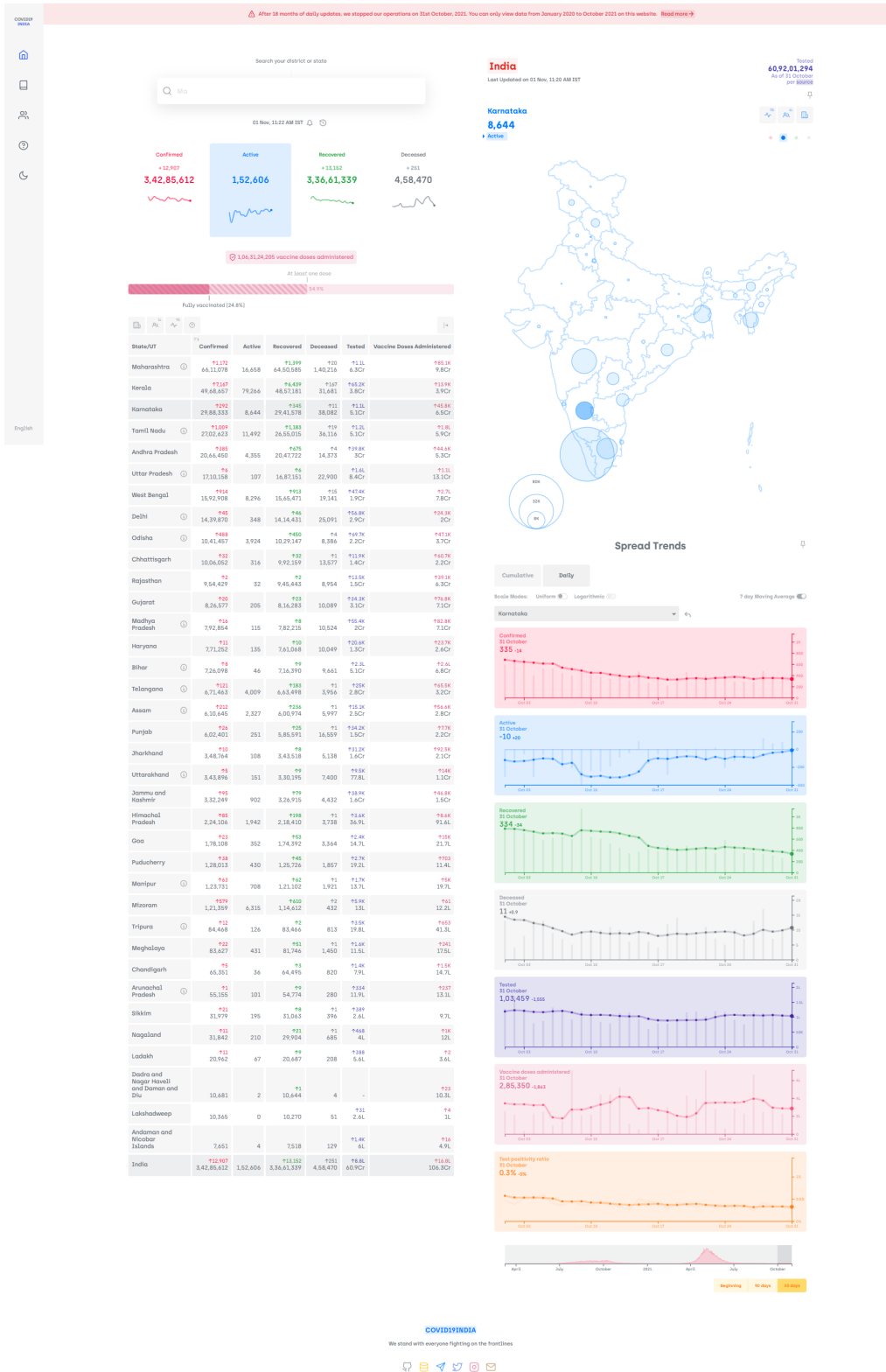


Figura 3.1: Página principal do COVID19INDIA [Covid19India, 2021].

gráficos de linhas e a tabela de valores.

- A **barra de pesquisa** permite pesquisar qualquer estado ou distrito do país. Carregar num dos resultados leva-nos para a página de distrito/estado, que é praticamente uma réplica da página inicial, mas com informação específica ao distrito/estado selecionado.
- Os *sparklines* representam a variação geral do números de casos de COVID, separando-os em quatro categorias - casos confirmados, ativos, recuperados, falecidos - de uma maneira simples e altamente condensada. É importante referir que clicar numa destas quatro categorias, transforma as outras visualizações presentes na página (figura 3.2).



Figura 3.2: Os gráficos *sparkline* do COVID19INDIA [Covid19India, 2021].

- O **mapa** do país faz parte de uma solução dinâmica de visualização geográfica. Conforme a métrica e unidade escolhida, a interface alterna entre mapa coroplético e mapa de símbolos proporcionais. Pairar com o rato sobre um distrito/estado muda as outras visualizações na página e selecionar um distrito/estado leva-nos para a sua página (figura 3.3).
- A **tabela de dados** mostra-nos os valores globais (i.e. desde que a pandemia começou) das mesmas categorias representadas no mapa do país (colunas) para todos os estados/distritos da Índia (linhas). É possível ordenar por qualquer coluna, incluindo por número de casos ativos, por nome do estado/distrito ou pelo seu tamanho da população (figura 3.4). Selecionar ou pairar com o rato sobre qualquer linha mostra a progressão dos valores nos gráficos de linhas, explicados abaixo.
- Os **gráficos de linhas** representam a variação dos valores referidos acima num qualquer intervalo de tempo (figura 3.5). Existem 3 botões no fundo, que permitem ao utilizador escolher os três intervalos mais utilizados ao analisar este tipo de dados: desde há 30 dias, 90 dias, ou desde o início da recolha de dados. Para além disso, é possível arrastar uma pequena janela para personalizar ainda mais o intervalo de tempo desejado (figura 3.6).

3.1.3 Forças e Fraquezas

Convém mencionar a qualidade da implementação do conceito de vistas ligadas (ver Secção 2.3.2 - *Linking and Brushing*) que esta plataforma possui. Quase todas



Figura 3.3: O mapa interativo do COVID19INDIA [Covid19India, 2021].

State/UT	Confirmed	Active	Recovered	Deceased	Tests
Maharashtra	↑1,172 66,11,078	16,658	↑1,399 64,50,585	↑20 1,40,216	↑1.1 6.3C
Kerala	↑7,167 49,68,657	79,266	↑6,439 48,57,181	↑167 31,681	↑65.2 3.8C
Karnataka	↑292 29,88,333	8,644	↑345 29,41,578	↑11 38,082	↑1.1 5.1C
Tamil Nadu	↑1,009 27,02,623	11,492	↑1,183 26,55,015	↑19 36,116	↑1.2 5.1C
Andhra Pradesh	↑385 20,66,450	4,355	↑675 20,47,722	↑4 14,373	↑39.8 3C
Uttar Pradesh	↑6 17,10,158	107	↑6 16,87,151	22,900	↑1.6 8.4C
West Bengal	↑914 15,92,908	8,296	↑913 15,65,471	↑15 19,141	↑47.4 1.9C
Delhi	↑45 14,39,870	348	↑46 14,14,431	25,091	↑56.8 2.9C

Figura 3.4: A tabela de dados da página inicial do COVID19INDIA [Covid19India, 2021].

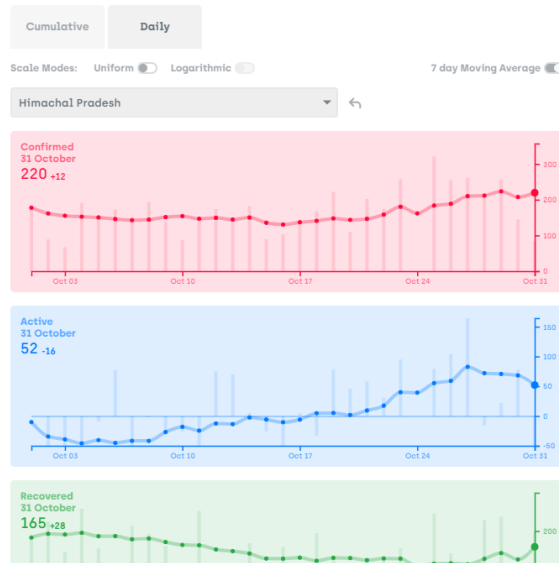


Figura 3.5: O conjunto de gráficos de linhas do *COVID19INDIA* [Covid19India, 2021].



Figura 3.6: Os controles de tempo dos gráficos de linhas do *COVID19INDIA* [Covid19India, 2021].

as interações (seleção de um local, de um intervalo de tempo, etc.) têm repercussões visuais no resto da plataforma, i.e. ajuda o utilizador a perceber como as várias visualizações se relacionam entre elas, permite obter um conhecimento mais holístico dos dados em questão, e possibilita a extração de conclusões de uma maneira intuitiva. Esta aplicação potencia grandemente a sua exploração, e capta perfeitamente a essência do discurso analítico - explorar, perguntar, obter respostas e visualizar correlações. Para além disso, a aplicação usa compreensivamente outras técnicas de interação mencionadas na secção 3.3, especialmente o *Overview plus Detail* - em elementos como o mapa geral do país *versus* o mapa do estado/distrito -, e o *Focus Plus Context*, que é aplicada, por exemplo, nos gráficos de linhas com o intervalo de tempo customizável. Por outro lado, não podemos deixar de apontar a redundância da tabela de dados que ocupa quase 40% das páginas principal. O utilizador comum não irá necessitar de analisar os dados concretos que alimentam as visualizações e parece-nos que cria ruído visual e complexidade a mais na interface.

3.2 Banda Ultra Larga Italia

O Plano Estratégico Banda Ultra Larga (ou simplesmente *BUL*)² - projeto a longo prazo onde a solução de visualização se insere -, tem como objetivo desenvolver uma rede de banda ultra-larga que se estende por todo o território nacional italiano. Com isto, o projeto almeja criar uma infra-estrutura pública de telecomunicações que coadune com as intenções da Agência Digital Europeia.

3.2.1 Objetivos

A plataforma que apoia este projeto tem como principal objetivo elucidar os utilizadores acerca do estado atual da implementação de soluções de telecomunicação (mais especificamente, fibra e *wireless*) nos vários concelhos e municípios que integram o território italiano. Estas soluções dividem-se em dois tipos, no que toca à entidade que as implementa: intervenção direta - quando ficam ao encargo da *Infratel* (uma empresa pública de *telecom*) e concessões (empresas privadas que disponibilizam o serviço de instalação das redes).

3.2.2 A Solução

A nível de interface, a aplicação apresenta dois elementos principais: o mapa coroplético (secção 3.1.3), e a *sidebar*.

- A plataforma tem como ponto focal um único tipo de mapa temático - o mapa coroplético (secção 3.1.3), e existem 3 vistas principais, que apenas diferem na granularidade geográfica com que se visualiza o território italiano - global, que inclui o país todo; distrito; e município. Na vista global, o utilizador consegue ver que distritos têm a maior taxa de redes completas implementadas, através da escala de azuis utilizada pelo mapa (figura 3.7). A nível distrital (figura 3.8), é possível visualizar o progresso da instalação das redes, segundo um código de cores: cinzento - planeamento completo; laranja e amarelo - design e arquitetura iniciados e completos, respetivamente -; escala de azuis - progresso da implementação física das redes (quanto mais escuro, maior o progresso). Na vista municipal, vemos as escolas, as casas, e as estradas, que já estão cobertas (ou não) pela rede (figura 3.9).
- Consoante a vista escolhida, o conteúdo mostrado na *sidebar* muda. A secção de progresso, onde a quantidade e estado de trabalhos de construção apenas é mostrada na vista distrital e global (figura 3.10). A secção de “Tecnologias oferecidas pelos operadores” também atualiza conforme o distrito selecionado, e mostra um gráfico empilhado horizontal que ilustra a quantidade de cidades com redes em planeamento, implementadas e em fase de testes (figura 3.11). Na vista municipal, podemos consultar uma *timeline*

²<https://bandaultralarga.italia.it/en/map/>

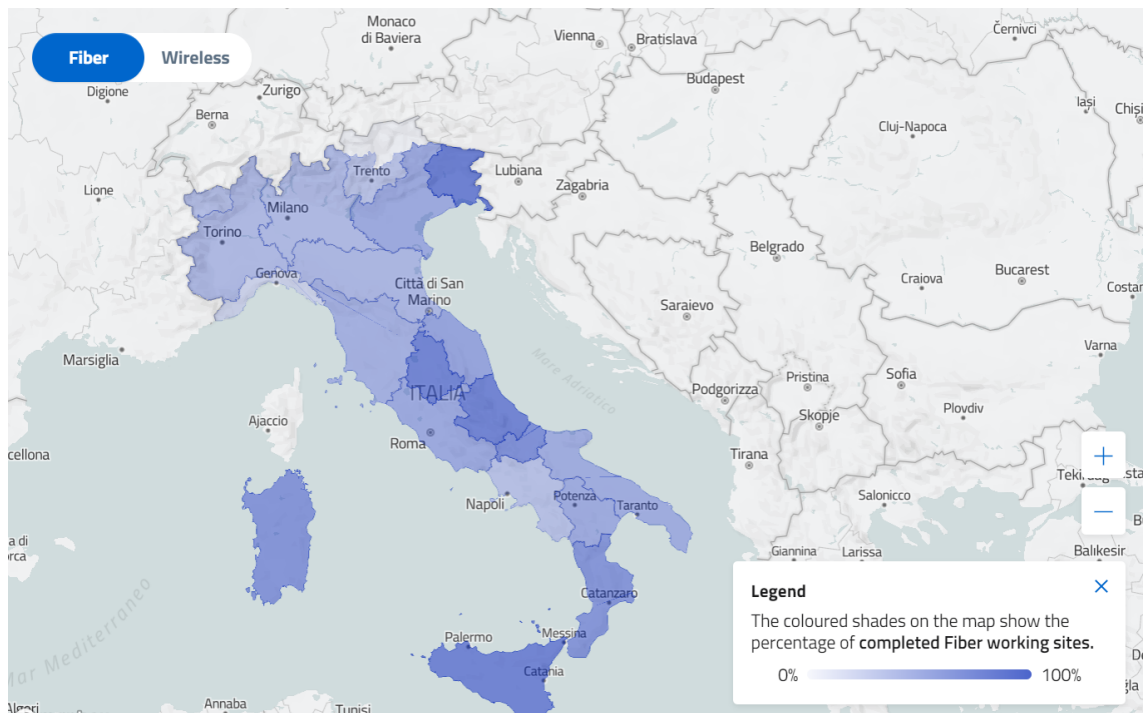


Figura 3.7: Vista global do BUL [BUL, 2020].

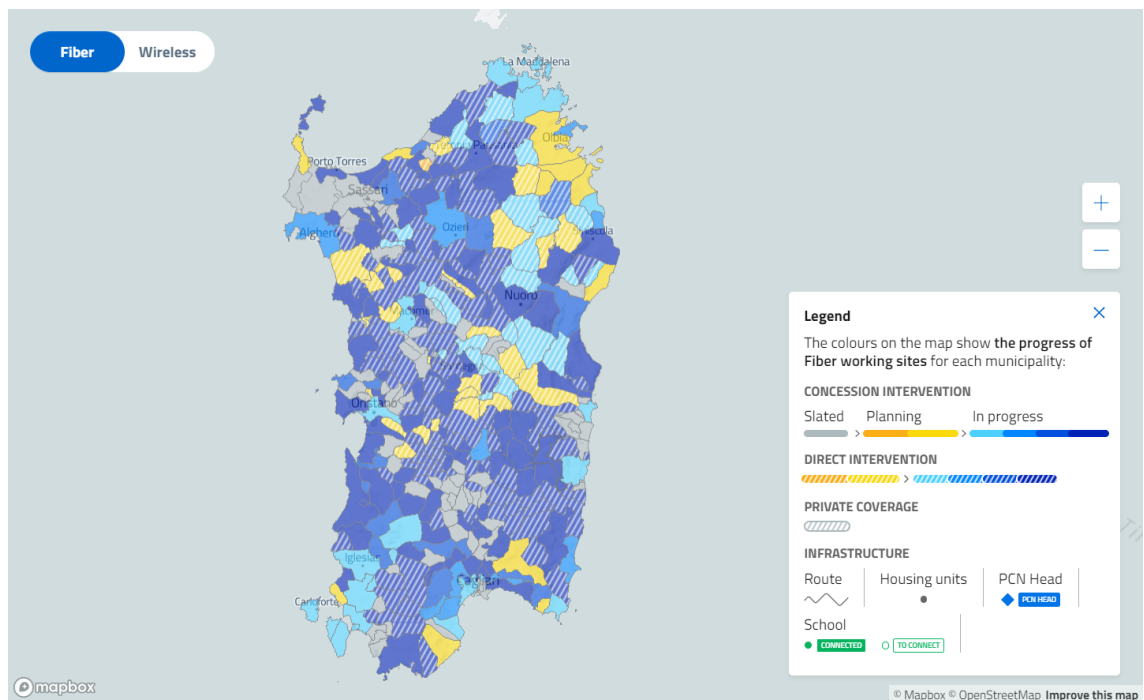


Figura 3.8: Vista distrital do BUL [BUL, 2020].

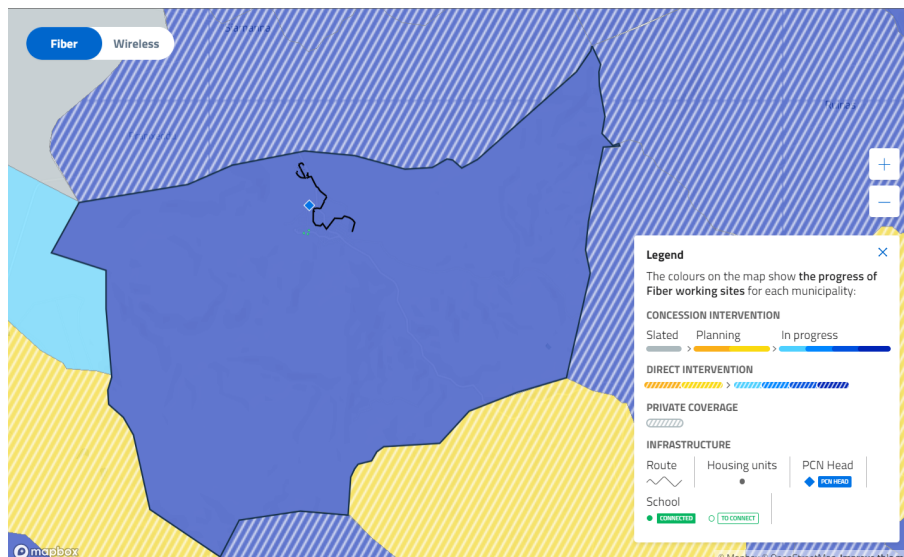


Figura 3.9: Vista municipal do BUL [BUL, 2020].

para saber o horizonte temporal do projeto numa dada localidade, inferindo assim se está atrasado ou adiantado relativamente ao previsto (figura 3.12).

3.2.3 Forças e Fraquezas

Descrita a plataforma, é de salientar a clareza com que a informação é codificada. Através de uma combinação de textura, cor e preenchimento das áreas geográficas relevantes o mapa é enriquecido com informação de uma maneira compacta e cristalina. No entanto, existem alguns pontos fracos que não podemos deixar de destacar.

Efetivamente, não é possível filtrar por áreas geográficas com uma certa característica comum. Um exemplo disto seria, por exemplo, na vista de distrito, mostrar apenas os municípios em fase de planeamento. Para além disso, é difícil comparar dois ou mais distritos ou municípios de forma clara e eficaz - por exemplo, apresentar dois distritos lado a lado para permitir a sua comparação direta.

3.3 UCLA Energy Atlas

O projeto *Energy Atlas*³, desenvolvido pelo *UCLA California Center for Sustainable Communities (CCSC)* da *UCLA*, no *Institute of the Environment and Sustainability (IOES)*, é uma base de dados do consumo de energia de edifícios que liga a informação das contas dos serviços públicos às características dos edifícios, dados sociodemográficos e outros atributos significativos que podem ser expressos espacialmente. A parte pública do projeto é uma plataforma de visualização que

³<https://energyatlas.ucla.edu/map>

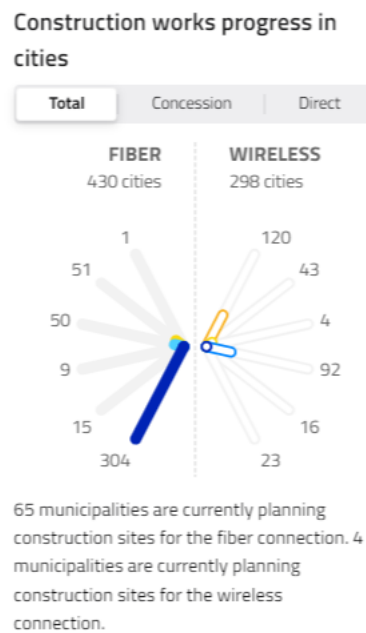


Figura 3.10: Quantidade e estado dos trabalhos de construção das redes do BUL [BUL, 2020].

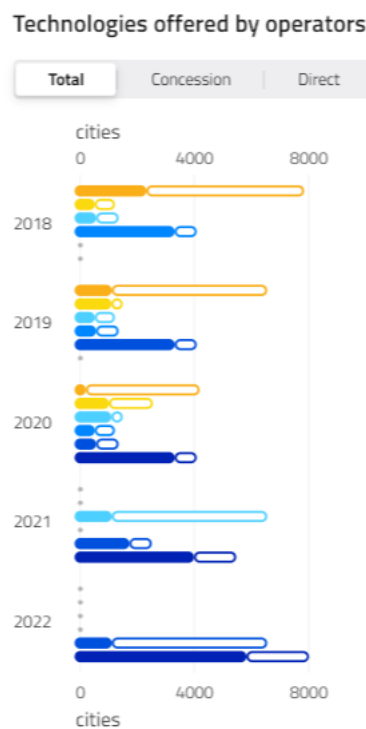


Figura 3.11: Tecnologias oferecidas pelos operadores do BUL [BUL, 2020].

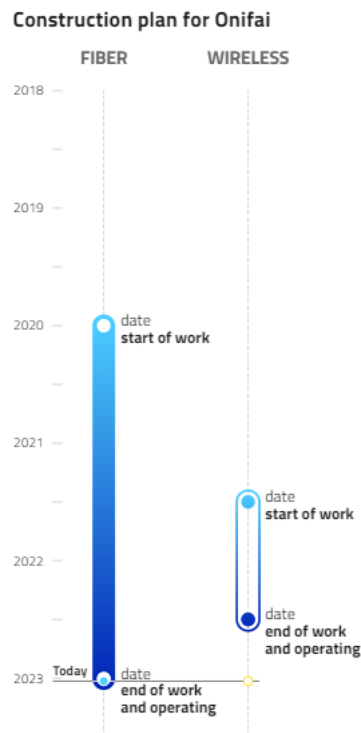


Figura 3.12: Horizonte temporal previsto (ou concretizado) da implementação das redes *BUL* [BUL, 2020].

exibe estatísticas de consumo de energia agregadas espacialmente com uma resolução temporal anual para a maioria dos bairros, cidades e condados no Sul da Califórnia [Pincetl et al., 2020].

3.3.1 Objetivos

Os edifícios são responsáveis por 40% das emissões de gases com efeito estufa (GEE) nos Estados Unidos. O projecto Energy Atlas proporciona aos californianos, e aos interessados nas transições de energia em curso, a oportunidade de interagir com um dos maiores conjuntos de dados desagregados de energia de edifícios disponíveis no país. Esta ferramenta interactiva é utilizada por uma grande variedade de grupos de interessados para informar o planeamento e a investigação no domínio da energia em toda a Califórnia [Pincetl et al., 2020].

3.3.2 A Solução

A interface principal desta aplicação subdivide-se em quatro elementos principais:

- **Mapa interactivo** - O mapa mostra visualmente o consumo de energia em todo o Condado, Conselhos de Governo do Condado de LA (COG), Cidade

e Bairro (apenas Condado de LA), bem como por tipo de edifício, idade, tipo de energia e emissões de gases com efeito de estufa. Os utilizadores podem visualizar dados por consumo total, consumo médio, média por metro quadrado de consumo e média *per capita* (residencial). Cada valor é calculado a partir de contas individuais e agregado para proteger a privacidade do cliente. Algumas áreas são omitidas para proteger a privacidade do cliente (figura 3.13).

- **Perfil** - Os perfis fornecem informação detalhada descarregável para cada um dos bairros, cidades e condados da Califórnia, incluindo dados longitudinais de 2006-2016, quando disponíveis (figura 3.14).
- **Data Stories** - As *data stories* analisam relações e tendências entre variáveis, incluindo o uso de energia e características dos edifícios, tais como idade dos edifícios, a sua área, tipo de uso, e características sociodemográficas.
- **Estratégias** - Esta secção fornece estratégias de conservação de energia e uma série de informações sobre políticas energéticas estatais e locais, incentivos e objectivos nos domínios da eficiência energética e das energias renováveis, bem como ligações a outros recursos nestes domínios.

3.3.3 Forças e Fraquezas

A utilização de um mapa como elemento de visualização primária é apropriada para a exibição de dados geográficos, e as diferentes cores utilizadas para representar diferentes níveis de utilização de energia são claramente diferenciadas e fáceis de interpretar. Além disso, a capacidade de ampliar e clicar em estados individuais para ver informação mais detalhada ajuda a proporcionar uma compreensão mais profunda dos dados. A inclusão de uma legenda para explicar a escala de cores também é útil. Globalmente, as visualizações no site são eficazes na comunicação dos dados e ajudam os utilizadores a compreender os padrões e tendências de utilização de energia em todo o país.

3.4 Síntese

Em conclusão, este capítulo visava analisar e dissecar algumas soluções existentes no campo da visualização geográfica. Descrevemos as plataformas, os problemas que resolvem e realçámos os seus pontos fortes e fracos. Em geral, vale a pena notar que todas estas soluções fornecem um bom ponto de partida e podem ser reutilizadas em trabalhos futuros, mas é importante considerar os pontos fortes e fracos mencionados, e fazer os ajustes necessários para que melhor se adequem às necessidades específicas do projecto.

Efetivamente, devemos mencionar que há duas técnicas de interação proeminentes em todas as aplicações - o *Linking and Brushing* e o *Overview plus Detail*. Ambas proporcionam formas mais interactivas e dinâmicas de explorar os dados. Esta é

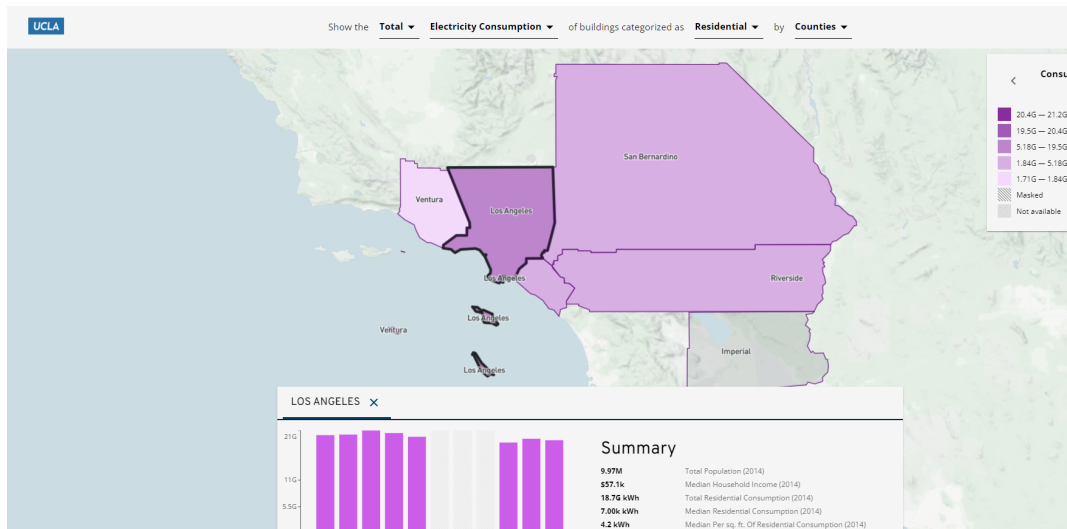


Figura 3.13: Captura de ecrã do mapa interativo do *UCLA Energy Atlas* [Pincetl et al., 2020].

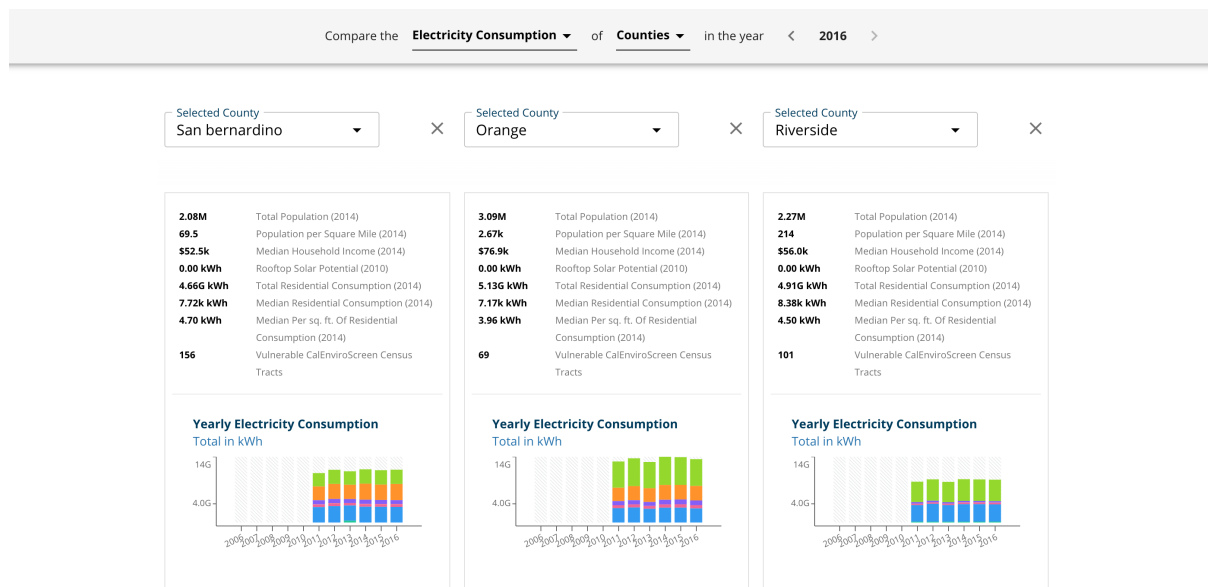


Figura 3.14: Captura de ecrã da interface de escolha de perfil do *UCLA Energy Atlas* [Pincetl et al., 2020].

uma característica valiosa, uma vez que permite aos utilizadores navegar e compreender facilmente os dados. Além disso, a capacidade de fornecer uma visão geral mais detalhada é benéfica, uma vez que potencia uma compreensão mais abrangente dos dados. Paralelamente, é interessante investir tempo considerável na codificação visual das variáveis, especialmente quando são em quantidades consideráveis. Todas as plataformas analisadas apresentam uma atenção redobrada a esse nível, e, portanto, é uma característica necessária à plataforma que será desenvolvida. Adicionalmente, concluímos que a filtragem por área geográfica é um atributo a desenvolver. Tendo tudo isto em conta, é importante lembrar que, querendo implementar vários tipos de visualizações e elementos gráficos que complementem a plataforma, não é desejável obstruir demasiado a vista do utilizador - o mesmo tem que sentir que a aplicação não está demasiado confusa e/ou desorganizada.

Em suma, o estudo destes casos de sucesso proporcionou uma visão aprofundada de como a implementação metódica dos conceitos básicos da visualização mudam a forma como o utilizador pode extrair conhecimento de uma forma mais (ou menos) eficaz. A análise das práticas e metodologias utilizadas nas diferentes plataformas estudadas permitiu estudar as melhores abordagens que contribuirão diretamente para o desenvolvimento deste projeto. Foi também possível ter a perceção dos desafios que podem surgir durante o processo de implementação e de que forma estes podem ser superados de modo eficiente.

Capítulo 4

Objetivos e Metodologias

Com este capítulo pretende-se fornecer uma visão global dos objetivos que se pretendem alcançar com esta dissertação, assim como os processos e métodos utilizados para o fazer.

4.0.1 Objetivos

O trabalho desta dissertação consistiu em desenvolver uma plataforma de visualização de dados climáticos que apoie a análise e tomada de decisões, utilizando uma arquitectura hierárquica de componentes *web*. Esta plataforma ambiciona dinamizar a integração de dados recém-recolhidos, assim com a sua posterior análise, de uma maneira que possibilite a extração de conclusões acerca de fenómenos climáticos desconhecidos e a confirmação de novos fenómenos do mesmo foro. Para o conseguir, o objectivo principal é desenvolver componentes de visualização avançados que constituirão a interface da aplicação, alavancando o conceito de modularidade que tão comum é no panorama atual do desenvolvimento *web*. Além disso, são estudadas técnicas de visualização de informação que ajudam na exploração, análise, e comunicação de dados atmosféricos, especificamente os que utilizam estruturas geo-temporais.

A primeira fase envolve a implementação de modelos de visualização existentes em componentes hierárquicos da *web* e a realização de explorações visuais iniciais. Na segunda fase, é desenvolvido um sistema de visualização combinando componentes básicos e modelos específicos para a visualização de dados atmosféricos. Os resultados são avaliados através de estudos empíricos e testes de utilizador.

Os objectivos supramencionados destinam-se a ajudar a garantir que a ferramenta a ser desenvolvida é eficaz na comunicação de informação aos seus utilizadores, e baseiam-se nos objectivos e necessidades desses utilizadores.

4.0.2 Metodologia

Nesta subsecção, detalhamos o método geral que planeamos utilizar para desenvolver a solução proposta acima. Baseámo-nos no modelo para design e validação de visualizações proposto por Munzner em 2009.

Esta metodologia geral para visualização de informação envolve uma série de passos sequenciais, dos quais destacamos os seguintes [Munzner, 2009] [Munzner, 2014]:

- *Domain problem and data characterization* - Este primeiro nível diz respeito à caracterização de tarefas e dados do utilizador final num dado domínio-alvo em particular. Isto envolve a compreensão do vocabulário do domínio, da maneira como os dados são utilizados, e dos desafios a serem resolvidos. Deste processo resulta um conjunto de questões colocadas ou acções realizadas pelos utilizadores-alvo;
- *Data type and operation abstraction design* - Esta camada envolve a tradução e o mapeamento do vocabulário de domínio para o vocabulário de visualização de informação. Isto é, descrição de operações (i.e, tarefas do domínio da visualização) e tipos de dados, que serão utilizado na codificação visual. No nosso caso, consiste em manipular a informação relativa ao locais de recolha das temperaturas, assim como as temperaturas em si, de maneira a que seja consumida pela interface de uma maneira eficaz e com boa performance.
- *Visual encoding and interaction design* - O terceiro nível consiste em codificar a informação e desenhar a interacção visual.
- *Algorithm design* - A última fase consiste no desenvolvimento de algoritmos a implementar as codificações visuais e interacções.

Capítulo 5

Trabalho Prático

Neste capítulo, descrevemos todos os tópicos relevantes ao trabalho prático que foi executado durante o segundo semestre. Discutimos os dados que nos foram disponibilizados e como foram modelados para permitir o máximo de flexibilidade e generabilidade; apresentamos a plataforma, sua arquitetura global e detalhes específicos de implementação; e, por fim, detalhamos o processo de validação da plataforma com recurso a testes de utilizador, assim como os seus resultados e ilações daí tiradas.

5.1 Planeamento e Gestão de Projeto

A gestão bem sucedida das várias componente que compõem uma plataforma deste género desempenha um papel fundamental no que toca a atingir os objectivos do projeto e na obtenção de resultados de elevada qualidade. Assim, é de especial importância abordar os aspectos cruciais da organização e execução eficazes das complexidades e particularidades deste projeto de uma forma sistemática e estruturada. Este capítulo explora os princípios fundamentais e as metodologias que guiaram o planeamento e execução deste projeto - desde estudo de requisitos e análise de risco até gestão de tarefas e de código -, mencionando também as tecnologias e ferramentas utilizadas para o efeito.

5.1.1 Requisitos e análise de risco

Nesta secção, descrevem-se as especificações e critérios fundamentais que orientaram o desenvolvimento da plataforma desenvolvida. O início deste projeto foi marcado por um conjunto de objectivos claros e bem definidos: criar uma plataforma que responda perfeitamente às necessidades dos utilizadores no terreno, permitindo-lhes registar instantaneamente as medições diretamente no sistema à medida que são recolhidas, e que, ao mesmo tempo, se transforme numa ferramenta robusta de análise a posteriori, facilitando a consulta, exploração e extração de conhecimentos valiosos dos dados acumulados. Para além disso, temos noção que o desenvolvimento de software abriga inerentemente uma panóplia

de incertezas que, inerentemente, dão origem a riscos. Neste sentido, inclui-se uma breve análise de riscos, onde se identificam e desenvolvem estratégias para a sua mitigação.

Requisitos funcionais

Os requisitos funcionais referem-se a descrições das características de uma plataforma e utilizam uma linguagem técnica para descrever. A priorização destes requisitos funcionais é crucial para discernir os aspectos vitais da plataforma dos menos significativos, de modo a que seja dado o devido ênfase a cada um durante o desenvolvimento. Para efetuar esta priorização, será utilizado o método MoSCoW [Haughey, 2011]. Esta abordagem consiste no seguinte:

- **Must (M):** Requisitos não negociáveis e obrigatórios para que a plataforma funcione como pretendido.
- **Should (S):** Requisitos importantes que acrescentam um valor significativo à plataforma, mas não são essenciais.
- **Could (C):** Requisitos que têm menor importância e podem não ser negociáveis se excluídos.
- **Won't (W):** Requisitos considerado sem importância ou sem valor dentro do período de tempo considerado que não serão implementados na iteração atual.

O inventário completo dos requisitos está documentado no Anexo A. No entanto, a Tabela 5.1 demonstra a estrutura dos requisitos funcionais.

Tabela 5.1: Requisitos funcionais da configuração da plataforma

Área	ID	Descrição	Prioridade
Upload de dados	CS-1	Adicionar uma variável nova à plataforma	M
	CS-2	Adicionar uma localização nova à plataforma	M
	CS-3	Adicionar uma medição nova à plataforma	M

Requisitos não-funcionais

Embora exista um largo consenso quanto à definição de requisitos funcionais, a definição de requisitos não-funcionais (RNFs) varia consoante a literatura consultada. No entanto, Kotonya e Sommerville [Kotonya and Sommerville, 1998] definem-nos da seguinte forma:

"Os requisitos não-funcionais são aqueles que não estão especificamente relacionados com a funcionalidade de um sistema, mas colocam restrições ao produto que está a ser desenvolvido e ao processo de desenvolvimento."

Tabela 5.2: Requisitos não-funcionais do desempenho da plataforma

	Descrição
Origem	Utilizador
Estímulo	Utilizador realiza um pedido ao <i>backend</i>
Resposta	<i>Backend</i> responde com sucesso
Métrica de validação	Tempo de resposta é menor que 1 segundo

Para além da sua prioridade (baixa, média, alta), os requisitos não-funcionais são descritos através dos seguintes atributos: origem, estímulo, cenário, resposta e métrica de validação.

Desempenho Este indicador descreve o desempenho do sistema quando responde a eventos com resultados. Como este sistema não é crítico no que respeita a operações sensíveis ao tempo, é-lhe atribuída uma prioridade média. No entanto, deve comportar-se de forma a que os utilizadores sintam que a navegação e a interação são contínuas e fluidas. Para tal, as respostas devem ser dadas num segundo ou menos. Este limite corresponde ao limite em que o fluxo de pensamento de um utilizador típico se mantém ininterrupto [Nielsen, 1994]. Idealmente, isto seria feito em menos de 0.1 segundos, uma vez que esse é o limite para o utilizador sentir que o sistema está a reagir instantaneamente.

Versatilidade A plataforma deverá ser adequada em dois *use-cases* específicos. O primeiro é a de ferramenta de terreno - os utilizadores poderão levar a ferramenta para os locais de medição e imediatamente registar na plataforma os valores medidos. A segunda função é a de ferramenta de análise a-posteriori - i.e. possibilitar a extração de conclusões-chave acerca dos dados, numa fase posterior à recolha. Este requisito será estabelecido como de alta prioridade, por se tratar de uma característica importante à utilidade da plataforma em vários contextos.

Usabilidade A usabilidade de um sistema pode ser vista na facilidade com que um utilizador pode aprender, navegar e utilizar o sistema para as tarefas pretendidas. Trata-se de uma característica importante para atrair novos utilizadores para a plataforma e mantê-los satisfeitos. Este requisito será estabelecido como de alta prioridade. O raciocínio subjacente a esta decisão é que, numa plataforma do tipo da que foi desenvolvida, uma boa usabilidade é chave pois afeta a funcionalidade e a utilização do sistema.

Dado que a usabilidade também engloba a facilidade com que um utilizador pode aprender o sistema, a secção 5.4. engloba testes de usabilidade com utilizadores que contactam com a plataforma pela primeira vez.

Tabela 5.3: Requisitos não-funcionais da versatilidade da plataforma

	Descrição
Origem	Utilizador
Estímulo	Utilizador insere dados na plataforma
Resposta	Dados são inseridos com sucesso
Métrica de validação	O dashboard é atualizado com os dados inseridos

Tabela 5.4: Requisitos não-funcionais da usabilidade da plataforma

	Descrição
Origem	Plataforma
Estímulo	Utilizador realiza os testes de usabilidade
Resposta	Questionário realizado pós teste de usabilidade
Métrica de validação	75% das respostas do questionário favorecem a plataforma

Análise de risco

Neste domínio, a análise de risco surge como um esforço vital, envolvendo a intrincada interação entre a avaliação da probabilidade de um evento específico se manifestar e a revelação das potenciais ramificações que ele pode acarretar. Este processo engloba três fases distintas: a identificação de possíveis riscos, uma avaliação da sua probabilidade e influência, e a formulação estratégica de medidas correctivas para reduzir o seu potencial impacto [Dey et al., 2007]. Essas contra-medidas abrangem um amplo leque de abordagens, que vão desde intervenções preventivas - que remodelam os próprios processos a partir dos quais os riscos emergem -, a medidas proactivas - que preparam contra riscos iminentes -, e respostas reactivas, que entram em ação assim que os riscos se materializam [Boehm, 1991].

A presente secção descreve os riscos identificados, e detalha as estratégias de mitigação desenvolvidas para cada risco. Os riscos são caracterizados pelas condições que podem provocar a sua ocorrência, as suas consequências e o seu plano de mitigação. Será desenvolvida uma matriz de riscos para visualizar a probabilidade de ocorrência e o potencial impacto caso ocorram.

Risco #1 - Falta de tempo para finalizar o projeto É pouco provável que este risco se concretize. Embora o desenvolvimento do projeto decorra em simultâneo com uma cadeira universitária durante o segundo semestre, o planeamento adequado continua garante a conclusão atempada dos objectivos do projeto sem atrasos. Para além disso, existe a possibilidade de uma entrega em setembro, durante a época especial. Este risco está detalhado na tabela 5.5.

Tabela 5.5: Descrição do risco #1

ID	1
Condição	Falta de tempo para finalizar o projeto.
Consequência	Impossibilidade de atingir os objectivos da dissertação, quer por incumprimento, quer por atraso.
Mitigação	Planeamento detalhado das actividades e tarefas a executar. O plano pode ser alterado em função da evolução das necessidades.

Risco #2 - Falta de capacidades técnicas na área de UI/UX É bastante provável que este risco se concretize, uma vez que o projeto presente será desenvolvido por um aluno com pouca (ou mesmo nenhuma) formação na área. Ainda assim, no caso de ocorrer, o seu impacto poderá ser mitigado pelo feedback do orientador ou de outros peritos na área e, em versões futuras, dos utilizadores. Este risco está descrito na tabela 5.6

Tabela 5.6: Descrição do risco #2

ID	2
Condição	Falta de capacidades técnicas na área de <i>UI/UX design</i> .
Consequência	A plataforma não cumpre os critérios de usabilidade e de interface do utilizador pretendidos. O produto final é incoerente e confuso para o utilizador.
Mitigação	Consultar peritos na área (nomeadamente o orientador), assegurando críticas informadas. Levar a cabo testes de utilizador para garantir que o produto se ajusta ao utilizador (e não o inverso), e assimilar o feedback do utilizador em versões futuras da plataforma.

É praticamente garantido que este risco se concretize. Os dados aos quais temos acesso foram recolhidos e armazenados por indivíduos sem formação em manipulação de dados, e portanto a probabilidade de haver falta de critério no pré-processamento da informação é relativamente grande. No entanto, temos uma grande variedade de soluções disponíveis para este problema específico. A descrição deste risco está presente na tabela 5.7

Tabela 5.7: Descrição do risco #3

ID	3
Condição	Baixa qualidade dos dados.
Consequência	Plataforma não tira máximo partido dos dados, levando a um produto final pouco convincente e possivelmente inútil.
Mitigação	Limpeza e pré-processamento minuciosos dos dados disponíveis. Adaptar a plataforma aos dados resultantes. Em último caso, estudar a possibilidade de utilizar o aumento de dados para mostrar o verdadeiro potencial da plataforma.

Risco #3 - Baixa qualidade dos dados

Risco #4 - Requisitos mal definidos Este risco pode gerar vários problemas, sendo o primeiro o atraso no desenvolvimento. Para além disso, pode levar a um produto final de qualidade inferior. É essencial compreender bem os requisitos e minimizar as alterações sempre que possível. As alterações aos requisitos podem revelar-se necessárias em fases posteriores. No entanto, estas inserirão inerentemente atrasos no desenvolvimento e devem ser evitadas através da recolha e validação cuidadosa dos requisitos nas fases iniciais. Este risco é detalhado na tabela 5.8

Tabela 5.8: Descrição do risco #4

ID	4
Condição	Requisitos mal definidos.
Consequência	Desenvolvimento incorreto ou desnecessário de funcionalidades, causando o seu incumprimento ou atraso. Alterações nos requisitos.
Mitigação	Rever cuidadosamente e iterar sobre os requisitos, garantindo um conjunto de restrições e necessidades coerente e suficiente para o desenvolvimento de um MVP.

Risco #5 - Requisitos mal prioritizados À semelhança do risco #4, este risco pode ser resultado de uma elicitação de requisitos inadequada, e pode concretizar-se, caso não se cumpra o plano de mitigação. Neste caso, pode ter implicações significativas para o projeto, uma vez que pode levar a um cenário em que se trabalha com os requisitos errados ou na ordem errada, que por sua vez, pode ter impacto na qualidade do produto final e na satisfação das partes interessadas. Assim, este risco pode também levar à ausência de funcionalidades cruciais nas primeiras versões. As suas especificidades estão descritas na tabela 5.9.

Tabela 5.9: Descrição do risco #5

ID	5
Condição	Requisitos mal priorizados.
Consequência	Incumprimento da implementação de funcionalidades importantes.
Mitigação	Organizar os requisitos com a ajuda de métodos de priorização de requisitos, como o modelo MoSCoW.

Matriz de risco As matrizes de risco podem ser utilizadas para agregar os riscos identificados, classificando-os em função da probabilidade da sua ocorrência e do seu impacto [Xiaosong et al., 2009]. A figura 5.4 é uma matriz de risco 3x3. Nesta, o eixo vertical representa a probabilidade de um risco acontecer, enquanto que o eixo horizontal representa o impacto do risco no projeto, caso aconteça.

Tabela 5.10: Classificação do impacto de um risco em categorias

Impacto	Descrição
Reduzido	Efeito nulo ou mínimo nos custos e no calendário. Os requisitos mínimos são cumpridos; a maioria dos requisitos secundários também o são.
Médio	Efeito tolerável no custo e no calendário. Os requisitos mínimos são cumpridos, porém os requisitos secundários podem não o ser.
Elevado	Efeito significativo ou crítico no custo e no calendário. Os requisitos mínimos podem não ser cumpridos.

Para o eixo do impacto, foram propostas três categorias, baseadas no trabalho de Xiaosong et al. [Xiaosong et al., 2009]. A tabela 5.10 apresenta as categorias propostas. O eixo da probabilidade de ocorrência foi também resumido em três categorias (tabela 5.11).

Tabela 5.11: Categorias de probabilidade

Probabilidade (%)	Categoria	Descrição
0 - 33	Reduzida	Improvável que aconteça.
34 - 66	Média	É igualmente provável que aconteça ou não.
67 - 100	Elevada	É provável ou muito provável que se concretize.

Com esta categorização em vigor, é agora possível traçar os riscos identificados numa matriz de risco. A matriz identifica a gravidade do risco, considerando

tanto a probabilidade como o impacto. Isto é representado com a ajuda de células coloridas e as etiquetas "Reduzido", "Médio", e "Elevado". Os riscos são identificados em cada uma das células correspondentes da matriz pelo seu ID.

Como é visível na figura 5.1, o risco #3 é o risco que merece mais atenção devido à sua elevada probabilidade e elevado impacto caso se concretize. Pode causar atrasos sérios no projeto se os dados estiverem num estado próximo do inutilizável.

Os riscos #4, #2 e #5 são de gravidade média. O risco #5, apesar do seu elevado impacto, tem uma baixa probabilidade de ocorrer devido à intensa revisão dos requisitos que deverá fazer com que quaisquer problemas sejam identificados atempadamente, assim como o risco #4. Para além disso, apesar de ter uma probabilidade média de se concretizar, o risco #2 poderá ser mitigado com relativa facilidade graças ao feedback de pessoas com o mínimo conhecimento de causa na matéria de UI/UX e portanto terá um impacto médio.

	Probabilidade →		
Impacto ↓	Reduzida	Média	Elevada
Reduzido			
Médio	1	4, 2	
Elevado	5		3

Figura 5.1: Matriz de risco

5.1.2 Gestão do progresso

Esta secção relata a forma e os recursos utilizados para levar a cabo o controlo do progresso, tanto em tarefas específicas, como no projeto em geral. São esclarecidas as metodologias relativas às tarefas que dos requisitos advieram, assim como aquelas relativas ao código implementado para concluir essas tarefas.

Gestão de tarefas

O acompanhamento do progresso do projeto em si foi feito com recurso a um quadro *Kanban*, com três colunas - *To-do*, *Doing*, e *Done*. Cada requisito é mapeado para uma tarefa (ou um conjunto de tarefas) específica(s), até estar completo. Cada tarefa poderia ter uma ou mais das seguintes etiquetas:

- *Data* - se a tarefa for relativa ao pré-processamento dos dados;
- *Backend* - se a tarefa for relativa às pesquisas, inserção ou remoção de itens na base de dados - i.e. qualquer operação que envolvesse a base de dados;

- *Frontend* - se a tarefa for relativa à interação do utilizador com a interface, lógica de componentes de visualização ou *design*;
- *Bug Fix* - se a tarefa for relativa à resolução de um *bug* encontrado numa iteração anterior do código durante testes.
- *Priority* - se a tarefa está no topo da lista de prioridades por ser bloqueante para outras tarefas.

A gestão destas tarefas com recurso ao quadro *Kanban* foi possibilitada pelo *Trello*, uma ferramenta de gestão de projetos que permite aos utilizadores criar quadros que representam diferentes projectos ou tarefas, que são depois divididos em listas e cartões personalizáveis. Cada cartão representa uma tarefa ou item específico e os utilizadores podem mover os cartões entre listas para indicar o seu progresso ou estado (figura 5.2)

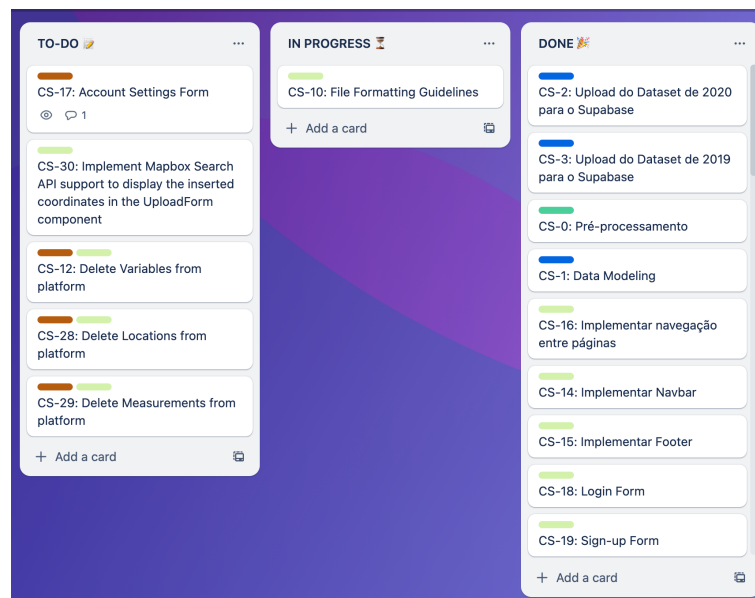


Figura 5.2: *Trello* - captura de ecrã do quadro *Kanban* do projeto durante o início do desenvolvimento. O código CS-XX (em que XX é um número qualquer) diz respeito à tarefa. Este código é único e não se repete, fazendo com que a tarefa seja o mais rastreável possível.

Gestão de código

À semelhança da forma como as tarefas do quadro *Kanban* eram um mapeamento próximo dos requisitos, também a gestão, controlo de versões e *tracking* do código desenvolvido para o projeto é um mapeamento das próprias tarefas. Para este efeito, a escolha foi o GitHub, que fornece aos programadores e às equipas de desenvolvimento um repositório centralizado para armazenar, gerir e acompanhar as alterações à sua *codebase*¹. O *lifecycle* de uma determinada funcionalidade tinha duas fases principais:

¹No centro do GitHub está o conceito de repositórios, que servem de contentores para projectos e o seu histórico de versões. Cada repositório contém um registo abrangente de todas

- Desenvolvimento do código correspondente num *feature branch*² novo;
- Abertura de um *pull request* para integrar as mudanças na *codebase* principal. Neste *pull request* são incluídos um título, uma breve descrição e o URL da tarefa correspondente (figura 5.3).

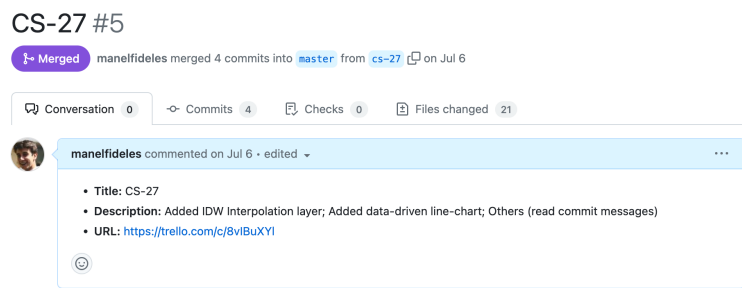


Figura 5.3: Captura de ecrã de um *pull request* para uma funcionalidade nova, incluindo o título, descrição e URL da tarefa correspondente. Repare-se que o código do *pull request* é igual ao código da tarefa correspondente.

Este *workflow* permite isolar as funcionalidades, e implementá-las uma a uma, mantendo um histórico de que tarefas foram executadas, quando foram começadas e quando foram finalizadas.

5.2 Dados

Nesta secção, referimos os dados ao nosso dispôr e que metodologia seguimos no seu pré-processamento, no seu armazenamento e gestão, as tecnologias usadas que possibilitaram a sua manipulação, assim como os desafios que foram ultrapassados e as dificuldades encontradas, nomeadamente a nível da flexibilidade da plataforma a vários tipos de dados climáticos.

5.2.1 Caracterização

O centro de investigação CEIS20 facultou-nos os dados necessários para a realização deste projeto. Assim sendo, os dados que temos ao nosso dispôr foram recolhidos por sensores climáticos instalados em drones. Estes drones são lançados de uma altitude padrão e, à medida que vão descendo em direção ao solo,

as alterações de código efectuadas pelos colaboradores, permitindo aos programadores reverter para versões anteriores, acompanhar o progresso e manter um histórico do desenvolvimento do projeto. O GitHub também oferece funcionalidades de ramificação, permitindo que os programadores trabalhem em novas funcionalidades ou correções de erros em ramificações, ou *branches*, separadas antes de as integrarem na *codebase* principal através de *pull requests*.

²No contexto do desenvolvimento de software, um *feature branch* é um ramo separado ou uma cópia da *codebase* principal que é criada para implementar uma funcionalidade específica ou trabalhar numa determinada tarefa. Permite que os programadores trabalhem em novas funcionalidades ou correções de erros isoladamente, sem afetar diretamente a *codebase* principal, muitas vezes referida como o *master* ou *main*.

vão medindo uma dada variável climática (p.e. a temperatura ou a humidade) a várias altitudes. No nosso caso específico, apenas temos acesso a dados relativos à temperatura, mas iremos abstrair-nos desse facto. Assumimos que qualquer variável climática é passível de ser medida com recurso a *drones*, desde que sejam equipados com sensores específicos para medição.

Assim, foram-nos facultados ficheiros que continham vários tipos de dados diferentes, sendo esses: a localização geográfica dos locais de recolha e as respetivas datas e horas de recolha; e as temperaturas medidas nesses locais e momentos no tempo. Os ficheiros, bem como os seus conteúdos, são os seguintes:

- *Levantamento 12-02-2019.xlsx* e *Levantamento 19-02-20.xlsx*, que contêm os nomes dos locais de recolha, assim como as suas respetivas coordenadas geográficas.
- *19-02-20.xlsx* e *12-02-19.xlsx*, que contêm as medições de temperatura para cada altitude (em graus célsius) em cada local de recolha.

A figura 5.4 mostra exemplos dos dados disponíveis. Do lado esquerdo da imagem (em cima e em baixo), temos os valores da variável climática medida (neste caso, a temperatura), em várias localizações, e as altitudes a que foram medidas. Do lado direito, em cima, temos parte dos nomes das localizações nas quais foram efetuadas medições, assim como as respetivas coordenadas. Por fim, no canto inferior direito, são apresentadas as horas e data de início e fim das medições, juntamente com o nome da localização nas quais foram efetuadas.

CS - 7h	CS - 15h	Altitude	LI - 7h45	Altitude
8.872	18.975	190.60	5.558	243.26
8.884	18.957	190.10	5.727	243.20
8.891	18.944	189.60	5.903	243.15
8.898	18.930	188.80	6.080	243.10
8.905	18.917	188.20	6.257	243.05
8.914	18.906	187.60	6.437	242.99

Local	X	Y
Linhares_1	40.22	-8.40
Casa_do_Sal	40.22	-8.44
Cemitério_Santo_An...	40.22	-8.40

Linhares 1	Linhares 2	Fucoli 1	Fucoli 2	Casa do Sal	Altitude_sonda
	8.158			9.935	188.91
	8.161			9.933	188.88
	8.166			9.930	188.80
	8.171			9.927	188.84
	8.175			9.925	188.84
8.07	8.179	9.759	9.681	9.925	188.12
8.07	8.187	9.764	9.688	9.925	187.10
8.08	8.194	9.770	9.694	9.924	186.26
8.09	8.200	9.777	9.703	9.924	186.18

Horário dos levantamentos - 12-02-2019			
Ordem	Local	Hora de Início	Hora de Fim
1	Casa do Sal	07:00:00	07:06:21
2	Linhares 1	08:00:00	08:09:05
3	Cemitério Santo António dos Olivais	08:30:00	08:35:12
4	Linhares 1	09:00:00	09:08:03
5	Rocha Nova	09:30:00	09:35:00
6	Linhares 1	15:00:00	15:08:03
7	Casa do Sal	15:17:00	15:23:21
8	Cemitério Santo António dos Olivais	16:30:00	16:35:12

Figura 5.4: Captura de ecrã de uma porção dos dados disponíveis. Para além dos atributos que necessitam de atenção, são visíveis as discrepâncias que existem no processamento pós-recolha dos dados - ficheiros diferentes que contêm o mesmo tipo de dados têm estruturas ou organizações diferentes, tornando o seu tratamento mais complexo.

Como podemos ver, existem vários atributos com os quais temos que lidar:

- O nome e localização geográfica dos **locais de recolha**, em pares de coordenadas (latitude, longitude);

- Datas e horas das **medições**;
- Os **valores** da variável climática medida - neste caso, a temperatura, em graus celsius -, nesses locais e momentos no tempo;
- A altitude (em metros) a que essa **variável** foi medida.

Como também se constata com relativa facilidade, os dados crus carecem de qualquer tipo de estrutura ou organização - ficheiros diferentes têm estruturas diferentes e com informação incompleta, duplicada ou irrelevante. Como já foi discutido, isto torna a sua consulta árdua e fastidiosa, e faz com que a confirmação do conhecimento existente e extração de novas conclusões sejam praticamente impossíveis.

5.2.2 Modelação

No sentido de garantir o máximo de flexibilidade a novas variáveis climáticas e tendo em conta os conteúdos dos ficheiros que nos foram fornecidos, começámos por identificar as entidades mais importantes presentes nos dados. Desta forma, foram identificadas quatro entidades cruciais:

- **Variáveis:** Que variáveis estão presentes na plataforma?
- **Localizações:** Que localizações estão presentes na plataforma?
- **Medições:** Quando e onde é que uma determinada variável foi medida?
- **Valores:** Quais os valores de uma variável nas várias altitudes a que foi medida, num determinado momento de medição?

De forma mais visual, a imagem 5.5 mostra a tradução dos items acima num diagrama ER ³. Assim, é fácil de perceber a divisão e separação das entidades e dos atributos que cada tabela da base de dados terá. Os atributos presentes nas entidades do diagrama ER são um mapeamento direto dos dados que nos foram disponibilizados. No entanto, é importante salientar que o atributo *display_color* da tabela *location*, não está presente em nenhum dos ficheiros fornecidos. Este atributo é utilizado para distinguir os valores presentes nas medições que correspondem às diferentes localizações, e é explicado em maior detalhe na secção 5.3.2 - *Dashboard*, onde é usado em vários elementos da interface.

Esta fase de planeamento e desenho é de enorme importância. Tendo definido os principais componentes da base de dados, o passo de pré-processamento torna-se mais simples, pois sabemos o aspeto que os dados terão que tomar após esse passo. Para além disso, é nesta fase que se encontram alguns erros de desenho,

³Um diagrama ER - ou diagrama Entidade-Relacionamento -, é uma representação visual utilizada no desenho de bases de dados para ilustrar as relações entre diferentes entidades que a compõem. Os diagramas ER são normalmente utilizados durante as fases iniciais do desenvolvimento da base de dados para ajudar as partes interessadas a compreender e definir o modelo de dados.

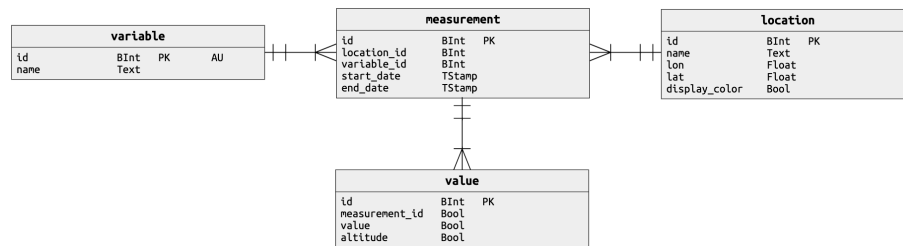


Figura 5.5: Diagrama ER da base de dados da plataforma - as quatro entidades cruciais e seus atributos.

que, se encontrados mais tarde no desenvolvimento do projeto, seriam caros de refazer ou mudar.

5.2.3 Pré-processamento e armazenamento

Como referido, os dados apresentam algumas inconsistências que necessitam de correção. A fase de pré-processamento consistiu nisso mesmo - remoção de valores duplicados e de informação incompleta. É importante relembrar que o nosso objetivo com este passo é manipular os dados de forma a que cumpram o diagrama apresentado anteriormente na figura 5.5. Deste processo resultaram os ficheiros *locations.csv*, *variables.csv*, *measurements.csv*, e *values.csv*, que replicam a estrutura de cada entidade correspondente no diagrama já mencionado. Para além disso, esta fase do projeto contribuiu também para que fossem elaboradas algumas regras que, do nosso ponto de vista, servem para facilitar a inserção de dados na plataforma, tanto para o utilizador - que não tem que se preocupar com formas de guardar os ficheiros -, como para a plataforma - que tem que verificar se os dados inseridos são válidos. Estas regras são listadas na secção 5.3.2 - *Settings*.

Após o pré-processamento dos dados, os resultados foram inseridos na base de dados. Graças às funcionalidades de *upload* de ficheiros CSV do Supabase - mencionado na secção 5.2.5 -, este passo foi extremamente simples. Nas figuras 5.6 a 5.9, vemos o resultados das operações descritas nesta secção. Foram também criadas réplicas de todas estas tabelas, que foram utilizadas durante desenvolvimento e em testes, evitando danos inadvertidos aos dados reais.

5.2.4 Dificuldades e desafios

O maior desafio desta fase do projeto foi a falta de homogeneidade dos ficheiros por processar. Medições em anos diferentes foram registadas de formas diferentes, o que significa que o método de pré-processamento teve que ser adaptado a cada ficheiro, tornando-se assim num processo moroso e de difícil execução. Para além disso, foi interessante desenhar uma estrutura de base de dados que garantisse a generabilidade e robustez da plataforma, para tornar possível a introdução de novas variáveis climáticas, medições e localizações na plataforma.

id	name	lon	lat	display_color
1	Linhares 1	-8.402028	40.222645	#8F0DA4
2	Pediátrico 1	-8.42241867112619	40.225139306405	#DEDD32
3	Fucoli 1	-8.42888798564339	40.2209505471073	#2B446E
4	Casa do Sal	-8.43730043314126	40.2191692143264	#948F78
5	Rocha Nova	-8.395038	40.237747	#F2844B
6	Linhares 2	-8.39957490384027	40.2247451644405	#4DF884
7	Linhares 3	-8.39727987464586	40.2228444234987	#E16462
8	Pediátrico 2	-8.42262686613226	40.2240673261461	#70FB51
9	Fucoli 2	-8.43018130754563	40.2221973437929	#900C00
10	Cemitério Santo Antó	-8.40355022347606	40.2182898879968	#4A58DD

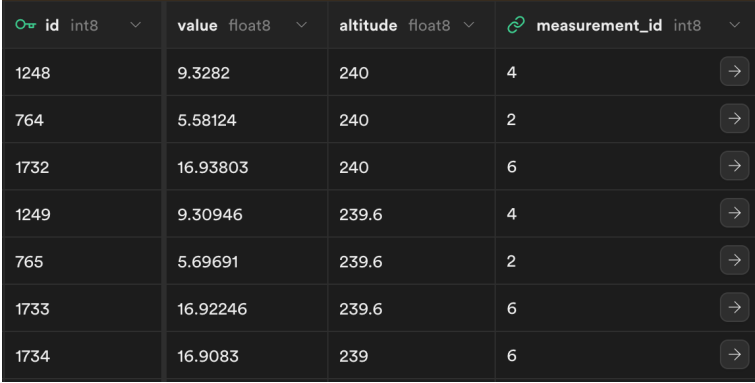
Figura 5.6: Captura de ecrã da tabela de localizações, após carregamento do respetivo CSV para o Supabase. Cada entrada nesta tabela caracteriza uma localização, representada pelo seu identificador único, nome, coordenadas geográficas e código hexadecimal de cor para facilitar a distinção dos dados correspondentes a cada localização.

id	location_id	start_date	end_date	variable_id
1	4	2019-02-12 07:00:00+00	2019-02-12 07:06:21+00	1
2	1	2019-02-12 08:00:00+00	2019-02-12 08:09:05+00	1
3	10	2019-02-12 08:30:00+00	2019-02-12 08:35:12+00	1
4	1	2019-02-12 09:00:00+00	2019-02-12 09:08:03+00	1
5	5	2019-02-12 09:30:00+00	2019-02-12 09:35:00+00	1
6	1	2019-02-12 15:00:00+00	2019-02-12 15:08:03+00	1

Figura 5.7: Captura de ecrã da tabela de medições, após carregamento do seu CSV para o Supabase. Cada entrada nesta tabela representa uma medição, caracterizada pelo seu identificador, o identificador da localização onde foi efetuada, datas e horas de início e fim, e o identificador da variável que foi medida.

id	name
1	temperature

Figura 5.8: Captura de ecrã da tabela de variáveis, após carregamento do seu CSV para o Supabase. Cada entrada nesta tabela representa uma variável, que é caracterizada apenas pelo seu identificador e nome.



id int8	value float8	altitude float8	measurement_id int8
1248	9.3282	240	4
764	5.58124	240	2
1732	16.93803	240	6
1249	9.30946	239.6	4
765	5.69691	239.6	2
1733	16.92246	239.6	6
1734	16.9083	239	6

Figura 5.9: Captura de ecrã da tabela de valores, após carregamento do seu CSV para o Supabase. Cada entrada nesta tabela é caracterizada pelo seu identificador, o valor registado a uma certa altitude, assim como o identificador da medição onde se insere.

5.2.5 Tecnologias utilizadas

A pilha de tecnologias utilizadas consistiu numa biblioteca para manipulação de dados, e num serviço para inserção e armazenamento da informação:

- **Pré-processamento** - Pandas. Esta biblioteca foi escolhida pelo facto de ser amplamente utilizada e otimizada para o tipo de *use-case* que temos em mãos, no qual lidamos estritamente com dados tabulares.
- **Base de dados e gestão dos dados** - Supabase. Este BaaS foi escolhido pelo facto de utilizar PostgreSQL - i.e. um paradigma relacional, mais adequado para lidar com os dados em formato de tabela aos quais tivemos acesso -; pelo facto de oferecer um SDK em JavaScript/TypeScript que facilita a integração com as restantes componentes arquitetónicas da plataforma (ver secção 5.3.1 – *Arquitetura*), e por tornar extremamente fácil o carregamento da informação para a base de dados através de um ficheiro CSV. A utilização de um BaaS retira bastante complexidade ao projeto, pois permitiu-nos focar apenas na componente de visualização e na interação do utilizador com a plataforma.

5.3 Plataforma

Nesta secção detalhamos como a solução desenvolvida está construída, dando primeiro uma vista global sobre o funcionamento geral, aprofundando gradualmente para uma vista particular, analisando cada página e seus componentes mais importantes. Para além disso, são elicitadas algumas dificuldades sentidas no desenvolvimento e implementação das várias componentes que integram a plataforma. Adicionalmente, são listadas as tecnologias utilizadas para tornar esta parte do projeto possível, juntamente com uma discussão acerca das razões pelas quais foram escolhidas.

5.3.1 Arquitetura

A arquitetura da plataforma requer uma explicação mínima para que as restantes secções relativas à ferramenta desenvolvida façam sentido, e para dar uma perspetiva mais holística do seu funcionamento.

A plataforma alavanca a arquitetura cliente-servidor, em que a aplicação cliente é desenvolvida em React (ver secção 5.3.4) que usa o Supabase como *backend-as-a-service* (ver secção 5.2.5).

O cliente React é responsável por renderizar a interface de utilizador, lidar com as interações dos utilizadores e fazer pedidos de dados e serviços ao *backend* do Supabase. Comunica com o *backend* através de pedidos HTTP, utilizando o SDK disponibilizado pelo Supabase, já mencionado na secção 5.2.5. A figura 5.10 mostra uma ilustração simplificada desta arquitetura.

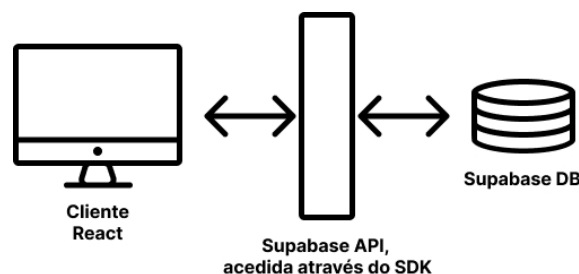


Figura 5.10: Representação simplificada da arquitetura Cliente-Servidor utilizada no projeto. O cliente React e o *backend* Supabase são unidos pela API do serviço, através do SDK providenciado.

É importante mencionar que todos os pedidos HTTP feitos ao *backend* são executados utilizando RPCs⁴, para os quais o Supabase oferece uma interface fácil de manipular. Esta filosofia baseada em RPC permite-nos reduzir o código necessário para efetuar pedidos ao *backend*, pois as funções invocadas estão localizadas no servidor. Para além disso, chamar uma função utilizando este método torna o código mais transparente e legível. Na figura 5.11 podemos ver o código necessário para invocar uma função no servidor utilizando os métodos disponibilizados pelo SDK do Supabase, a partir do cliente; e na figura 5.12 podemos ver a função correspondente que é executada no servidor.

```
const { data: searchData, isLoading } = useRPC({
  rpcName: 'search_locations_by_string',
  params: { search_term: searchInputValue },
});
```

Figura 5.11: Captura de ecrã do código necessário para invocar uma função no servidor através de RPC. Neste caso, as instruções na figura chamam a função *search_locations_by_string* com o argumento *search_term*.

⁴Um Remote Procedure Call, ou RPC, é um protocolo que permite a um programa executar funções num servidor ou serviço remoto como se fossem locais para o cliente

```
CREATE OR REPLACE FUNCTION search_locations_by_string(search_term text)
RETURNS TABLE (name text, id integer, lat float, lon float)
AS $$
    SELECT l.name, l.id, l.lat, l.lon
    FROM locations l
    WHERE l.name ILIKE '% ' || search_term || ' %';
$$ LANGUAGE sql;
```

Figura 5.12: Captura de ecrã da função invocada na figura 5.11. Esta função SQL é executada no lado do servidor após ser chamada pelo cliente, fazendo uma *query* sobre todas as localizações disponíveis na plataforma e retornando uma cujo nome corresponda ao parâmetro *search_term* enviado pelo *frontend*.

No geral, esta arquitetura é uma combinação adequada ao problema e ao âmbito do projeto, permitindo-nos focar no desenvolvimento da interface do utilizador e na lógica *client-side*, abstraindo a implementação do lado do servidor - que é assegurada pelo Supabase -, em funcionalidades como autenticação, operações sobre os dados (p.e. consulta), entre outros.

5.3.2 Páginas

Nesta secção, elicitamos as várias páginas que compõem a plataforma, desvendando a sua finalidade, lógica de implementação e interação no contexto da ferramenta desenvolvida. A plataforma implementa as seguintes páginas: *Register* e *Login*, onde o utilizador se inscreve e entra na plataforma; *Settings*, que permite controlar os dados que são mostrados na plataforma, assim como a alteração de alguns dados pessoais; e *Dashboard*, que é página central da plataforma. Nesta página, o utilizador tem acesso à informação disponível, e pode manipulá-la, filtrá-la e visualizá-la à sua maneira.

Register/Login

Embora estas páginas sejam menos relevantes no âmbito do projeto, não deixam de ser importantes para garantir a coerência e completude da plataforma.

A página de registo (figura 5.13) inclui quatro campos para introdução de dados pelo utilizador - o primeiro nome, o último nome, o e-mail, e a palavra-passe. O utilizador pode ainda esconder ou mostrar a palavra passe, caso a queira retificar. Ao clicar no botão *Sign up*, é feito um pedido ao *backend* que cria um perfil de utilizador com os dados inseridos e se for bem sucedido, devolve, entre outras informações, um *access_token* e um *user_id*. Estes dados são essenciais para verificar se o utilizador tem acesso aos dados apresentados no *dashboard*.



Sign In [Sign Up](#)

Sign up
to leverage the full power of Calescope 🔥

First Name * Last Name *

Email *

Password *

[Sign up](#)

Already a user? [Login](#)

Figura 5.13: Captura de ecrã da página de registo.

A página de início de sessão (figura 5.14) - i.e. *login* -, tem praticamente o mesmo fluxo. Apresenta dois campos - um email e uma palavra-passe -, e na submissão, é enviado um pedido ao *backend* que devolve a mesma informação que na página de registo. É importante lembrar que este fluxo de autenticação do lado do servidor é-nos (quase) totalmente opaco, por estarmos a utilizar o SDK do Supabase, que abstrai toda a complexidade ao processo - p.e. não precisamos de gerar *access_tokens* ou inserir utilizadores na base de dados.



Sign In [Sign Up](#)

Sign in to your account

Email *

manueelfideles77@gmail.com

Password *

.....

[Sign in](#)

Don't have an account yet? [Register here.](#)

Figura 5.14: Captura de ecrã da página de início de sessão.

Settings

O objetivo e foco desta página é duplo - permitir ao utilizador alterar a sua informação pessoal - i.e. primeiro e último nome, e sistema de unidades (Sistema Internacional ou Imperial)⁵ -; e configurar as variáveis climáticas presentes na plataforma.

Ao carregar no botão *Edit* no topo da página, o utilizador desbloqueia os campos de *input* para os poder mudar - por defeito, estes campos estão bloqueados para permitir que o utilizador veja os seus dados sem os alterar por engano (figura 5.15). Tendo acesso a esses dados, o utilizador pode alterá-los como desejar. Se estiver satisfeito, pode carregar no botão *Save*. Inversamente, se desejar reverter as alterações efetuadas, basta carregar no botão *Cancel* (figura 5.16).

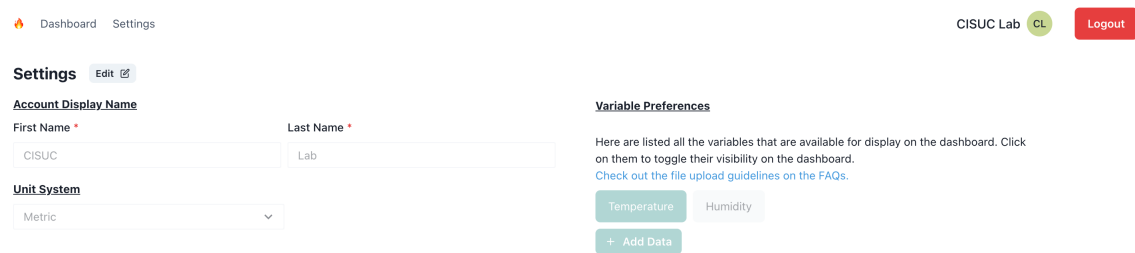


Figura 5.15: Captura de ecrã da página de *Settings* fora do modo de edição - Os campos de *input* estão todos bloqueados, permitindo que o utilizador consulte as suas definições sem as alterar por engano.

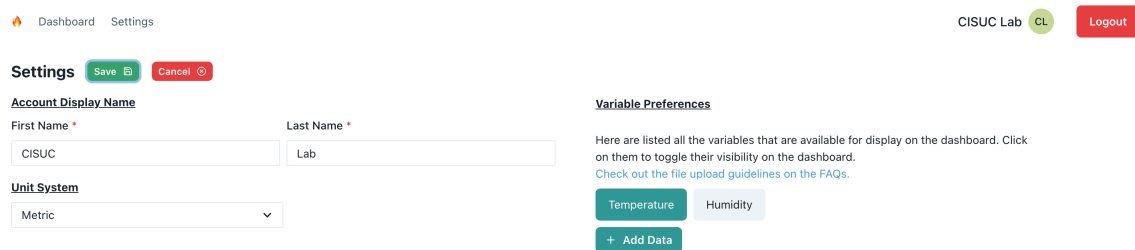


Figura 5.16: Captura de ecrã da página de *Settings* no modo de edição - Os campos de *input* estão ativos, permitindo que o utilizador altere as suas definições.

A configuração dos dados pessoais é bastante simples - são apenas campos de texto que o utilizador usa (figura 5.16, do lado esquerdo) para alterar a sua informação. Do lado direito desta página, são apresentadas todas as variáveis climáticas que o utilizador já inseriu na plataforma - no caso das figuras 5.15 e 5.16, são apresentadas *Temperature* e *Humidity*. Carregando numa qualquer (no modo de edição) alterna a sua visibilidade no *dashboard* - se o retângulo correspondente estiver verde, a variável será visível no *dashboard*, caso contrário, estará escondida. Esta funcionalidade é mencionada na secção 5.3.2 - *Dashboard*.

⁵Fluxos secundários, como mudar a palavra-passe ou alterar o e-mail, não fazem parte dos nossos requisitos por estarem demasiado fora do âmbito da plataforma, então não foram implementados.

Ainda em relação a esta página, o botão *Add Data* ativa um modal⁶ que permite ao utilizador adicionar novas localizações, variáveis climáticas e medições, através do carregamento de ficheiros CSV.

Este modal é composto por três passos distintos:

1. **Location** - O utilizador pode escolher uma das localizações já guardadas na plataforma ou criar uma nova. Se optar pela primeira opção, os campos *Latitude* e *Longitude*, completam-se automaticamente (figura 5.17). Caso queira criar um local novo, é possível fazê-lo simplesmente escrevendo no campo *Name* o nome que se pretende dar ao local - p.e. Nova Iorque -, e seleccionar a opção *Create* que é mostrada - neste caso *Create "Nova Iorque"* (figura 5.18). Naturalmente, escolhendo esta opção, o utilizador tem que inserir as coordenadas geográficas do local novo. Para além disso, do lado direito deste passo, é apresentado um pequeno mapa que se centra automaticamente nas coordenadas do local, assim que o utilizador insere as insere. Desta forma, o utilizador consegue verificar se as coordenadas que inseriu estão corretas.

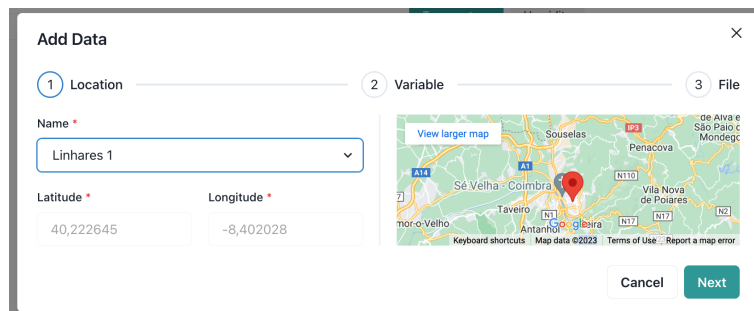


Figura 5.17: Seleção de um local de medição pré-existente na plataforma. Ao seleccionar um local, os campos referentes às coordenadas geográficas preenchem-se automaticamente.

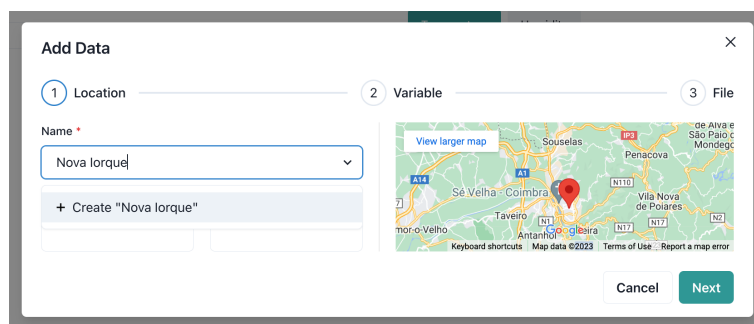


Figura 5.18: Criação de um local de medição novo. Ao digitar um nome de um local que não está guardado na plataforma, é apresentada a opção de criar um novo.

⁶No contexto do desenvolvimento web, um "modal" refere-se a um elemento da interface gráfica que interrompe temporariamente o conteúdo principal de uma página web ou aplicação para apresentar informações suplementares, recolher dados ou solicitar a interação do utilizador. Os modais são frequentemente apresentados como uma sobreposição flutuante sobre o conteúdo existente, escurecendo ou esbatendo o fundo para chamar a atenção para o conteúdo do modal.

2. **Variable** - Este passo é composto por duas componentes: do lado esquerdo, é possível inserir as datas e horas de início e fim do período de tempo no qual a medição foi efetuada - nos campos *Start Timestamp* e *End Timestamp*, respetivamente; e do lado direito, o utilizador escolhe a variável que foi medida nesse período de tempo. De forma semelhante à escolha da localização no primeiro passo, o utilizador pode escolher uma variável já guardada na plataforma (figura 5.19), ou simplesmente escrever o nome que quer dar a uma variável nova e selecionar a opção *Create* (figura 5.20).

Figura 5.19: Seleção de uma variável climática pré-existente na plataforma.

Figura 5.20: Criação de uma variável climática nova. Ao digitar um nome de uma variável que não está guardada na plataforma, é apresentada a opção de criar uma nova.

3. **File** - O último passo é constituído pelo carregamento dos valores correspondentes à localização, variável, e período de tempo de medição inseridos nos passos anteriores. Do lado esquerdo, é possível efetuar o *upload* do ficheiro CSV com os valores desejados, onde é mostrado o nome do ficheiro carregado (figura 5.21, do lado esquerdo). Do lado direito, podemos ver as primeiras linhas do ficheiro carregado para a plataforma - até 5 linhas (figura 5.21, do lado direito). É importante referir que o ficheiro carregado apenas pode ter duas colunas: a primeira coluna contém os valores da variável climática para cada altitude, e a segunda coluna contém os valores da altitude (figura 5.22)⁷. Para além disso, é efetuado um *parsing* básico do ficheiro assim que é carregado, verificando se o número de colunas nele presente é válido e eliminando entradas inutilizáveis - p.e. linhas do ficheiro que têm um valor de altitude, mas não têm um valor da variável (ou vice-versa).

⁷A plataforma assume que os dados presentes no ficheiro CSV carregado seguem o Sistema Internacional de unidades.

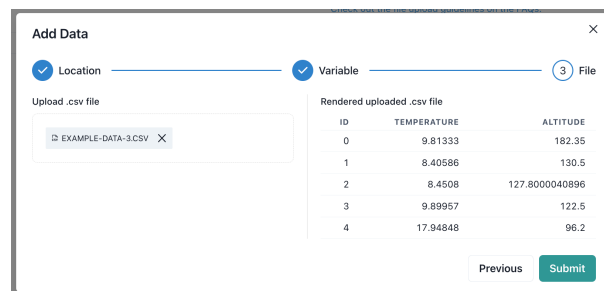


Figura 5.21: Carregamento de um ficheiro CSV com os valores correspondentes à localização, variável climática e período de tempo de medição previamente inseridos. Do lado esquerdo, é mostrado o nome do ficheiro carregado e, do lado direito, as suas primeiras linhas.

```
data > example-data > example-data-3.csv
1 9.44061,217.9
2 9.81333,182.35
3 8.40586,130.5
4 8.4508,127.8000040896
5 9.89957,122.5
6 17.94848,96.2
7 18.21096,84.7000027104
8 9.3945,83.9
9 19.7683,71.6000026112
10 9.96135,64.5
11 7.80138,62.490909090909085
12 10.47922,59.7000022304
13 20.34636,52.6000020032
14 10.75698,37.3000015136
15 3.86466,17.6000008832
16
```

Figura 5.22: Exemplo do formato do ficheiro CSV que a plataforma aceita como válido. A primeira coluna corresponde aos valores da variável climática medida, e a segunda coluna contém as altitudes a que a variável foi medida.

No final deste processo, o utilizador pode carregar em *Submit* para submeter os dados mencionados na plataforma. Se todos os campos estiverem preenchidos corretamente, a plataforma dá feedback positivo ao utilizador através de um *toast*⁸ com a mensagem "*Data uploaded successfully.*". Caso contrário - p.e. se o ficheiro carregado não for válido -, a plataforma dá feedback negativo com uma mensagem descritiva do erro.

Dashboard

O *dashboard* é constituído por apenas cinco componentes, elicitados nesta secção:

1. Mapa
2. Barra de pesquisa
3. Barra de filtros
4. Barra de detalhes
5. Controlos da camada de IDW

O *layout* da página está apresentado na figura 5.23, com cada um dos componentes numerados de 1 a 5, correspondendo à ordem da lista supramencionada.

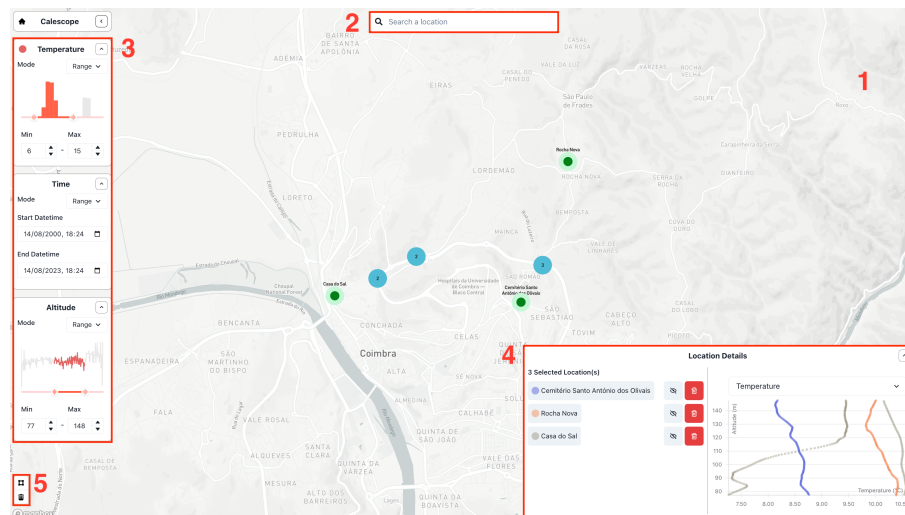


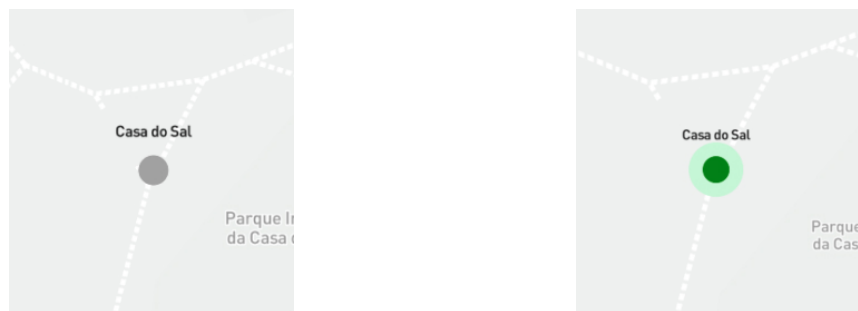
Figura 5.23: Captura de ecrã do *dashboard*. Esta página apresenta cinco elementos cruciais que interagem entre eles para compôr a ecrã principal da plataforma.

⁸Os *toasts* providenciam ao utilizador feedback não-intrusivo sobre o resultado de uma ação realizada ou para transmitir atualizações relevantes. Aparecem frequentemente como pequenas caixas retangulares perto da parte superior ou inferior do ecrã, com uma mensagem de texto curta e, por vezes, um ícone. Normalmente, os *toasts* são automaticamente encerrados após uma determinada duração, garantindo que não obstruem a vista do utilizador durante um período prolongado. Os casos de utilização comuns dos *toasts* incluem notificar os utilizadores de ações bem sucedidas (p.e. "Mensagem enviada"), alertá-los para erros (p.e. "Palavra-passe inválida"), ou apresentar mensagens de estado (p.e. "A carregar...").

Nesta secção descrevem-se as funcionalidades e papéis de cada um, detalhando a maneira como funcionam, interagem uns com os outros e possibilitam a exploração dos dados registados na plataforma.

Mapa O mapa é responsável por duas funções principais, e é o elemento que cobre maior área da página.

A primeira função é mostrar todas as localizações presentes na plataforma e permitir a sua seleção. As localizações são representadas como pontos no mapa. Se o nível de *zoom* no mapa for baixo, os pontos são agregados em *clusters* - representados por círculos com cor variável consoante o número de localizações que agregam (entre 2 e 4 é colorido a azul, de 4 a 9 é colorido a amarelo, e mais que 10 é vermelho), para facilitar a sua visualização e navegação no mapa. É possível selecionar uma localização no mapa através de um clique em cima do ponto representativo da mesma (figura 5.24a, do lado esquerdo). Ao fazê-lo, a localização mostra um estado de seleção, marcado por um círculo verde (figura 5.24b, do lado direito), ao mesmo tempo que é apresentado ao utilizador um *toast* que mostra que a localização clicada foi selecionada.



(a) Uma localização por seleccionar

(b) Uma localização selecionada

Figura 5.24: Capturas de ecrã dos dois estados de seleção possíveis para uma localização no mapa.

A segunda função do mapa é renderizar o resultado dos cálculos da interpolação de valores através de *Inverse Distance Weighting* (IDW). O IDW [Bartier and Keller, 1996] é um método de interpolação espacial utilizado em geoestatística e sistemas de informação geográfica (em inglês, *geographic information systems*, ou GIS) para estimar valores em locais não observados com base em valores observados em locais conhecidos próximos. Esta técnica assume que os valores numa determinada localização são mais influenciados por pontos próximos do que por pontos distantes.

Em termos simples, o IDW calcula uma média ponderada de valores conhecidos com base nas suas distâncias da localização alvo, dando mais peso aos pontos mais próximos e menos peso aos pontos mais afastados. A fórmula para IDW pode ser expressa como:

$$Z_p = \frac{\sum_{i=1}^n w_i \cdot Z_i}{\sum_{i=1}^n w_i} \quad (5.1)$$

Na qual:

- Z_p é o valor estimado na localização alvo.
- Z_i é o valor observado na localização conhecida i .
- w_i é o peso atribuído à localização conhecida i , calculado com base na sua distância à localização alvo.
- n é o número total de localizações conhecidas.

O cálculo dos pesos é feito com recurso a uma função matemática que atribui pesos maiores a pontos mais próximos e pesos menores a pontos mais distantes. No nosso caso, utilizamos uma função cúbica e não reparámos em nenhum problema de performance, pois esta técnica é teoricamente simples e computacionalmente eficiente. Contudo, para conjuntos de dados maiores, utilizar uma função quadrática poderá diminuir o tempo de computação e melhorar a performance. Para além disso, a proximidade dos valores das medições torna o IDW uma técnica adequada ao problema.

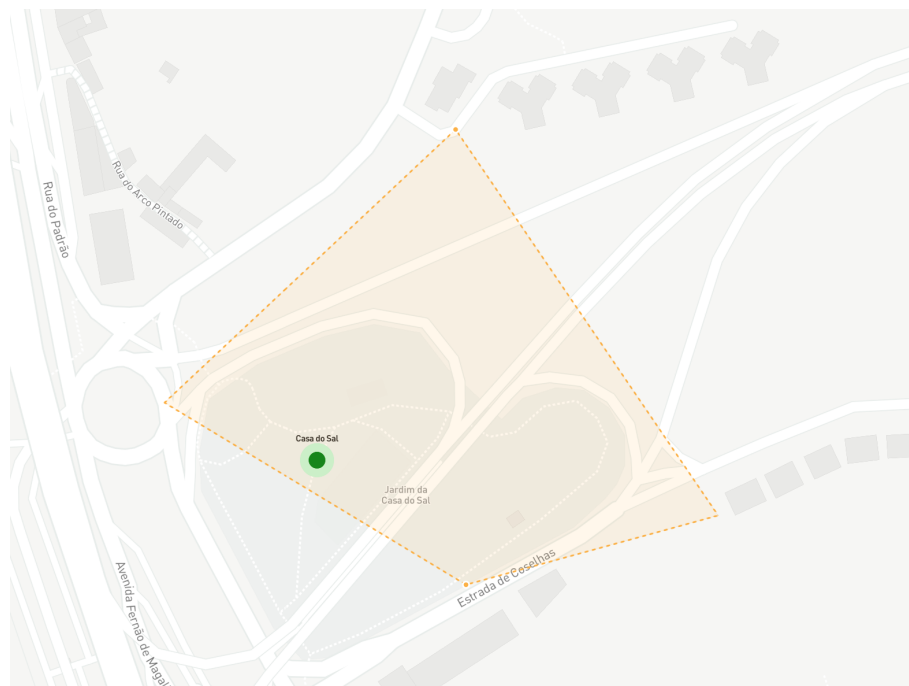


Figura 5.25: Captura de ecrã de um utilizador a desenhar a área de interpolação. Quando a área desejada é acabada de desenhar, a camada de interpolação é renderizada sobre o mapa.

Assim, o IDW gera uma superfície de interpolação, na qual os valores mudam gradualmente no espaço. No caso do *dashboard*, a renderização desta superfície é controlada pelo utilizador, permitindo-lhe que desenhe a área geográfica sobre a qual pretende aplicar a interpolação. Selecionando o modo de desenho, o utilizador consegue desenhar uma qualquer forma geométrica que englobe a área

geográfica desejada para interpolação (figura 5.25). Assim que o utilizador completa o desenho da forma, a interpolação é calculada e renderizada sobre o mapa (figura 5.26).

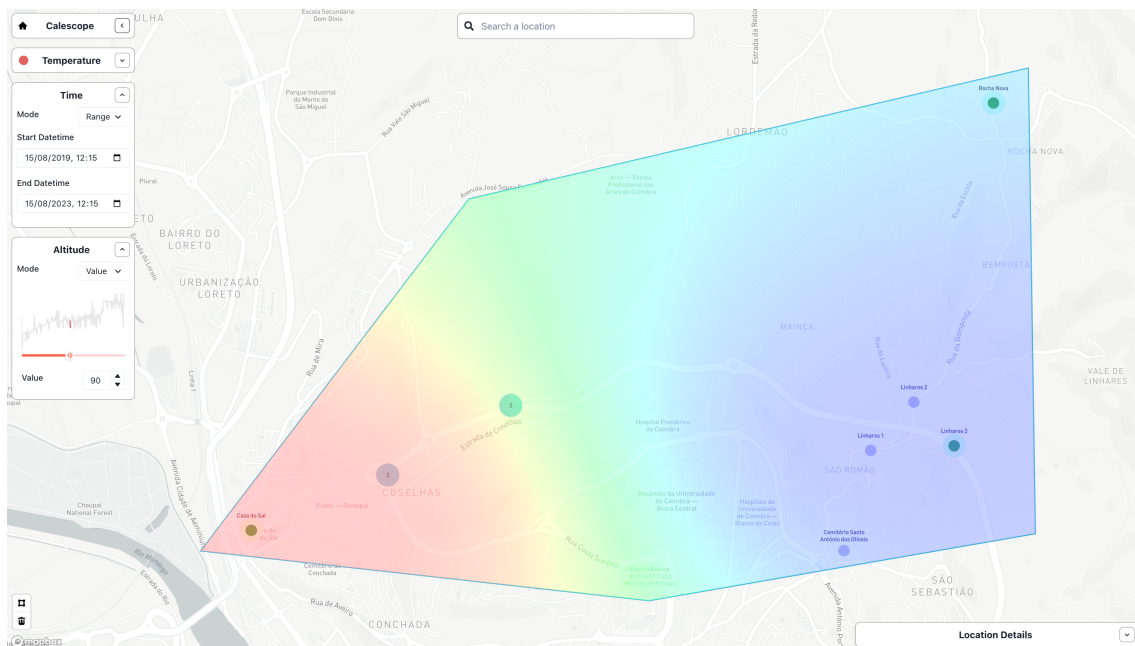


Figura 5.26: Captura de ecrã do resultado da interpolação com IDW. Esta funcionalidade utiliza os valores medidos a uma altitude definida nas localizações selecionadas para calcular e renderizar uma superfície de interpolação. Os valores mais elevados da variável em análise (neste caso a temperatura) são representados com a cor vermelha e os mais baixos com a cor azul.

Depois de criada, é possível manipular a superfície de interpolação através dos controlos localizados no canto inferior esquerdo. Este componente é composto por dois botões: o botão de desenho, que permite criar e editar a superfície à qual a interpolação é aplicada; e o botão de apagar, que permite eliminar a superfície de interpolação criada. Estes controlos são visíveis na figura 5.23, no canto inferior esquerdo numeradas com o número 5, após a seleção de duas ou mais localizações ou se já existir uma superfície criada.

Barra de pesquisa A barra de pesquisa, localizada na parte superior da página, permite pesquisar por nome as localizações já carregadas para a plataforma. É um componente bastante simples, mas que facilita a procura de localizações no mapa sem ser preciso navegar pelo mapa - sendo particularmente útil se o número de localizações for bastante elevado.

Os resultados da pesquisa são mostrados à medida que o utilizador digita uma dada cadeia de caracteres (figura 5.27). É também importante salientar que, se uma dada localização já estiver selecionada aquando do uso da barra de pesquisa, este estado é mostrado ao utilizador mudando a cor do ícone do resultado para verde (figura 5.27). Ao selecionar um resultado da pesquisa, o mapa é centrado nas coordenadas da localização correspondente (figura 5.33).

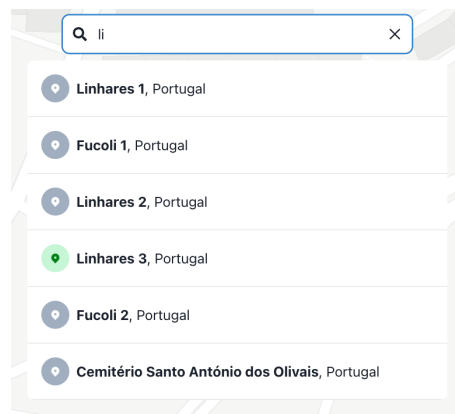


Figura 5.27: Captura de ecrã da barra de pesquisa durante uma pesquisa. Os resultados de pesquisa são atualizados enquanto o utilizador escreve. Para além disso, as localizações selecionadas são representadas com um ícone a verde, e as por seleccionar são representadas por um ícone cinzento.

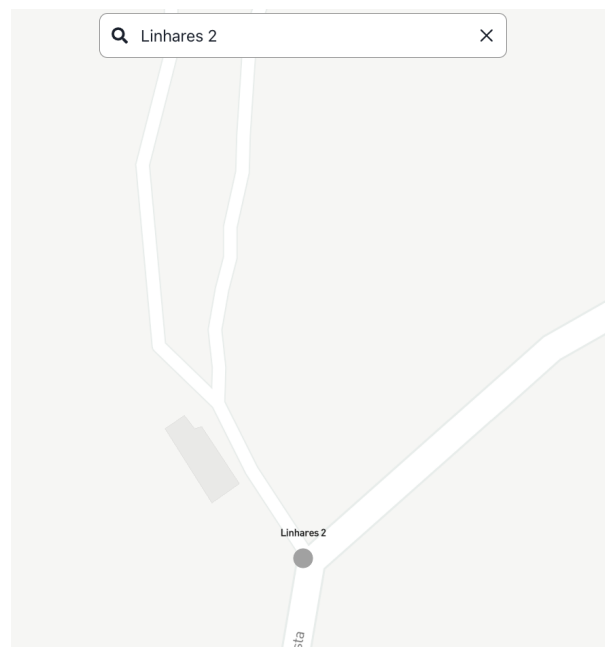


Figura 5.28: Captura de ecrã da barra de pesquisa a seguir a seleccionar um dos resultados. Após escolher num dos resultados apresentados, o mapa é centrado na localização clicada.

Barra de filtros A *sidebar* agrega todos os filtros da plataforma. Mostra as variáveis disponíveis para filtragem - definidas na página de *Settings* (secção 5.3.2), associando uma cor a cada uma delas, e é constituída por um conjunto de cartões colapsáveis (figura 5.29), onde cada cartão é referente a um filtro.

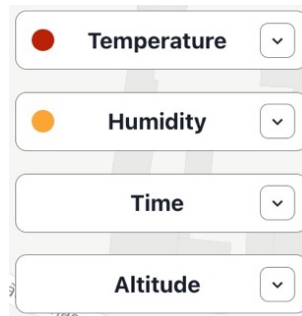
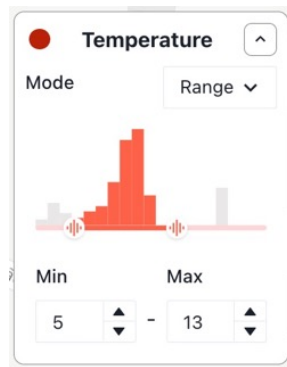


Figura 5.29: Captura de ecrã dos filtros de variáveis. Cada variável tem uma cor para que haja a sua fácil distinção no gráfico de linhas que é renderizado no filtro de altitude.

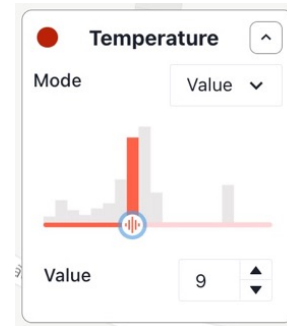
Os filtros são divididos em duas categorias gerais - os filtros de variáveis, e os filtros estáticos. Como o nome indica, os filtros de variáveis controlam a filtragem das variáveis que estiverem visíveis (secção 5.3.2). Os filtros estáticos são compostos pelo filtro da altitude e pelo filtro do tempo. Estes filtros são fixos, tanto pelos dados que temos (secção 5.2.2) como pelos requisitos do projeto - i.e. são sempre apresentados na interface do *dashboard*, independentemente das variáveis que estiverem visíveis. Tal como esperado, o filtro de altitude filtra as medições por altitude, e o filtro de tempo filtra as medições por data e hora em que foram feitas.

Todos os filtros - de variável ou estático -, apresentam dois modos diferentes de seleccionar valores - um modo "value" que permite seleccionar apenas um valor inteiro, e o modo "range", que permite seleccionar um intervalo de valores. Em modo "value", é apresentado apenas um *slider*⁹ com apenas um indicador e um campo de *input*, para seleccionar um valor específico. No modo "range", é apresentado um *slider* com dois indicadores, que permitem a seleção de um intervalo, arrastando os indicadores para os valores mínimos e máximos desejados.

⁹Um *slider* um elemento de controlo gráfico com o qual o utilizador pode definir um valor movendo um indicador, normalmente na horizontal. Em alguns casos, o utilizador pode também clicar num ponto do cursor para alterar a definição.

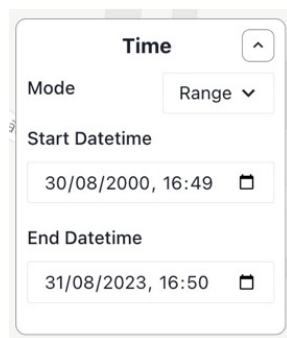


(a) Modo "range".

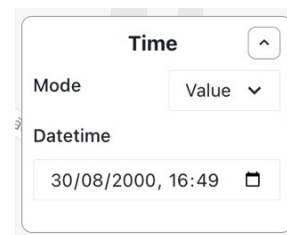


(b) Modo "value".

Figura 5.30: Capturas de ecrã dos dois modos possíveis de filtragem de valores dos filtros de variável e altitude.



(a) Modo "range".



(b) Modo "value".

Figura 5.31: Capturas de ecrã dos dois modos possíveis de filtragem de valores do filtro de tempo.

No caso específico do filtro de tempo, tanto o modo "range" como o modo "value" não apresentam *slider*.

Apesar dos pontos em comum, estes dois tipos de filtros apresentam as suas diferenças a nível das visualizações que renderizam. Para uma dada variável - p.e. a temperatura -, é renderizado um gráfico de barras, em que cada barra corresponde a um valor específico de temperatura, e a altura relativa da barra representa a quantidade de medições que têm esse valor de temperatura (figura 5.30a). Este gráfico é renderizado para qualquer variável climática em questão. Acerca dos filtros estáticos, o filtro de tempo não apresenta qualquer visualização, apenas o(s) campo(s) de *input*, como já mencionado acima. No caso do filtro da altitude, é renderizado um gráfico de linhas que apresenta a média dos valores de uma ou mais variáveis, para cada valor de altitude, em que cada linha corresponde a uma variável, codificada com a cor respetiva (figura 5.32).

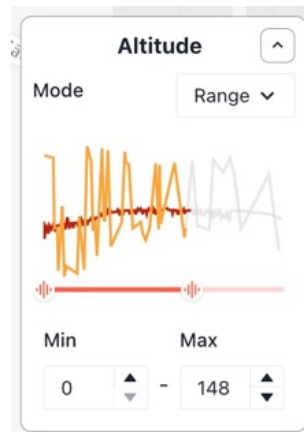


Figura 5.32: Captura de ecrã do filtro de altitude. Cada variável tem uma cor para que haja a sua fácil distinção no gráfico de linhas que é renderizado. Estas cores correspondem às cores na figura 5.29.

Barra de detalhes A barra de detalhes, localizada no canto inferior direito da página, tem duas secções: a listagem das localizações selecionadas e traçar o gráfico de pontos que diz respeito às medições efetuadas nas localizações selecionadas. Este componente altera-se automaticamente em função de duas ações - escolha de localidades através da sua seleção no mapa e mudanças a parâmetros da barra de filtros. A figura 5.33 mostra um exemplo da utilização da barra de detalhes.

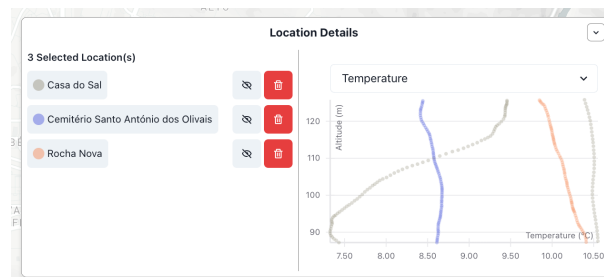


Figura 5.33: Captura de ecrã da barra de detalhes após seleção de três localidades. Do lado esquerdo, é apresentada uma lista das localidades selecionadas. Do lado direito, o gráfico de pontos com os dados relativos às medições

A barra de detalhes permite visualizar as localizações selecionadas em formato de lista. Nessa lista, é possível alternar a visibilidade das medições correspondentes a uma dada localidade no gráfico de pontos do lado direito. Adicionalmente, é também possível remover localizações da lista (figura 5.33, do lado esquerdo). Como já foi mencionado na secção 5.2.2, cada localização tem uma cor associada, para que haja distinção visual entre os valores das medições de cada uma.

Do lado direito do componente, o gráfico de pontos mostra os valores resultantes da aplicação dos filtros supramencionados (ver secção 5.3.2), para cada localização selecionada. As cores que representam cada localização (do lado esquerdo do nome da localização) são as mesmas que são utilizadas na lista de localizações

selecionadas. Nos casos em que existem medições em momentos no tempo diferentes para a mesma localização, passar com o rato por cima de uma qualquer série de pontos revela, no canto inferior esquerdo do gráfico, as datas e horas de início e fim da medição (figura 5.34).

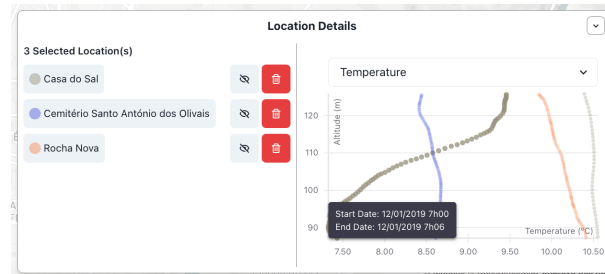


Figura 5.34: Captura de ecrã da barra de detalhes com medições diferentes para a mesma localidade. Na imagem, vemos que a localização *Casa do Sal* tem dois momentos de medição diferentes, o que pode ser confuso para o utilizador. Assim, ao passar com o rato por cima de qualquer uma delas, são reveladas, no canto inferior esquerdo do gráfico, as datas e horas de início e fim dos momentos de medição.

Por cima do gráfico de pontos, há ainda um menu que possibilita a troca da variável visualizada no gráfico de pontos. As variáveis disponíveis para visualização neste gráfico correspondem às variáveis que o utilizador define como visíveis na página *Settings* (secção 5.3.2)

5.3.3 Dificuldades e desafios

A maior dificuldade desta componente do projeto foi a ligação dos filtros ao resto da interface. Fazer com que toda a interface mudasse em função do *input* do utilizador, com os valores certos e consoante o modo de filtragem escolhido (*range* vs *value*), foi um processo moroso e de difícil implementação.

É preciso também mencionar que implementar a renderização do IDW sobre o mapa foi árduo - desde ponderar fazer a implementação da funcionalidade de raiz até à integração de bibliotecas com APIs difíceis de entender e manipular, como acabou por ser o caso.

Para além disso, dada a complexidade da interface em si, garantir que o *dashboard* se mantinha responsivo e com boa performance - mesmo com conjuntos de dados relativamente grandes -, provou-se um problema interessante e desafiante. Uma das soluções implementadas para solucionar grande parte das consequências desta questão foi o uso de memoização de componentes¹⁰, permitindo

¹⁰No React, a memoização é utilizada para otimizar o desempenho de um componente funcional memorizando o resultado de uma computação. Esta técnica ajuda a evitar re-computações desnecessárias de valores que não mudam entre as renderizações do componente. Assim, a sua utilização pode ser particularmente vantajosa quando se trata de cálculos dispendiosos ou de cálculos que não precisam de ser recalculados em cada renderização.

reduzir o número de renderizações necessárias para atualizar os dados visualizados. Igualmente, recorreremos ao uso de *debouncing*¹¹, que possibilita a diminuição do número de pedidos feitos ao *backend* quando o utilizador, num curto espaço de tempo, interage muitas vezes com componentes que com ele comunicam. Esta técnica reduz a carga sobre o servidor, que de outra maneira teria que fazer mais operações sobre os dados e responder a mais pedidos no mesmo espaço de tempo.

5.3.4 Tecnologias utilizadas

Dada a complexidade desta fase do projeto, a pilha de tecnologias é relativamente grande. As *frameworks*, bibliotecas e *packages* utilizadas foram escolhidas conforme as restrições de tempo e de capacidades técnicas abordadas na secção 5.1.3 - *Análise de risco*. Algumas das tecnologias abaixo são abordadas na secção 2.5 - *Tecnologias*, e portanto, a título de brevidade, não carecem de explicação adicional.

- **TypeScript:** Linguagem de programação desenvolvida pela Microsoft baseada na sintaxe do JavaScript, mas que introduz a verificação estática de tipos e outros recursos avançados. É frequentemente referida como um "super-conjunto" do JavaScript, o que significa que qualquer código válido escrito em JavaScript válido também é código válido em TypeScript. No entanto, o TypeScript acrescenta sintaxe e funcionalidades adicionais que permitem aos programadores escrever código mais estruturado, sustentável e resistente a erros. Através da implementação de tipagem, esta linguagem facilita a deteção atempada de erros e a escrita de código com maior qualidade, fornecendo uma estrutura clara e bem definida que o código escrito em JavaScript não possui por defeito.
- **React.js**
- **Chakra UI:** Biblioteca de componentes *open-source* desenvolvida com o objetivo de facilitar a criação de interfaces de utilizador em aplicações React. Esta tecnologia proporciona um conjunto de componentes de UI personalizáveis pré-feitos que simplificam o processo de desenvolvimento de UI, oferecendo um conjunto consistente e coerente de componentes, estilos e ferramentas.
- **React Map GL:** O *react-map-gl* é um conjunto de componentes React para bibliotecas Mapbox GL JS. É um *wrapper* que abstrai as complexidades de trabalhar com Mapbox GL JS diretamente.

¹¹O *debouncing* é uma técnica utilizada para controlar o ritmo de execução de uma determinada ação ou função. Geralmente, é utilizada em cenários em que eventos rápidos e frequentes, como *input* do utilizador ou eventos do *browser*, acionam uma função que pode ter operações que consomem muitos recursos ou interações com sistemas externos. Em termos simples, o *debouncing* garante que uma função só é invocada após um determinado período de inatividade, em vez de ser acionada imediatamente após cada evento. Isto ajuda a evitar que a função seja chamada com demasiada frequência, o que pode levar a cálculos desnecessários ou a problemas de desempenho.

- *D3*
- **React Vis:** React-vis é uma biblioteca de visualização em React construída sobre o D3.js, e é basicamente um *wrapper* React para o D3. Os componentes React-vis são projetados para funcionar como outros componentes React. É possível criar gráficos complexos com uma quantidade mínima de código. É também possível personalizar todos os aspectos de uma determinada visualização. Para além disso, esta biblioteca inclui vários tipos diferentes de gráficos pré-feitos, que aceleram o desenvolvimento de uma plataforma como a desenvolvida.

5.4 Validação

Nesta secção descrevemos a metodologia utilizada para validar a plataforma, assim como as conclusões tiradas do processo, referindo as várias fases dos testes de utilizador que foram levados a cabo.

Esta validação engloba, não apenas a avaliação da experiência de utilização da plataforma, mas também a verificação da eficiência dos vários componentes da plataforma e das suas funcionalidades. O objetivo é garantir que a interface seja intuitiva, fácil de utilizar e que atenda aos requisitos dos utilizadores, além de assegurar a consistência e a coerência da ferramenta em diferentes cenários de utilização. A validação será realizada por meio de testes rigorosos, envolvendo cinco utilizadores reais e aplicando métricas objetivas para avaliar a eficácia da plataforma e a satisfação dos utilizadores. Por fim, é feita uma reflexão crítica sobre os resultados do processo e sobre o feedback que nos foi dado pelos utilizadores submetidos ao processo de teste.

5.4.1 Metodologia

A validação da plataforma foi desempenhada sob a forma de testes de validação, compreendendo três fases distintas. O processo teve início com uma introdução ao trabalho desenvolvido e uma explicação das diversas fases do teste, seguida da assinatura de um documento de consentimento informado (anexo B.1).

Antes dos testes propriamente ditos, foram feitas algumas questões que permitiram recolher informações pessoais e demográficas de forma a poder ligar o perfil de cada participante aos resultados obtidos. Consiste em perguntas de cariz pessoal (idade, ocupação profissional, nível de escolaridade, capacidade informática, etc.), para adquirirmos uma noção das características básicas da pessoa testada. O questionário está disponível no anexo B.2, assim como as respostas dos sujeitos de teste ao questionário.

Após a recolha de dados procedeu-se à realização de tarefas com recurso ao MVP da plataforma que foi desenvolvido. Com o intuito de abranger múltiplos ecrãs e funcionalidades durante o processo de teste, foi solicitado aos sujeitos de teste que executassem um conjunto de tarefas relevantes na plataforma, em um ou vá-

rios cenários normais (i.e. tarefas que seriam expectáveis em contexto de pesquisa e investigação científica, ou durante mera exploração da plataforma). Assim, foram estabelecidas quatro tarefas que representam quatro fluxos importantes da plataforma¹²:

1. Procurar localizações
2. Filtrar valores
3. Aplicar IDW
4. Fazer *upload* de um ficheiro CSV.

Por fim, a usabilidade da interface foi avaliada com recurso à ferramenta *System Usability Scale* (SUS). O *SUS* consiste num questionário pós-teste com dez itens e uma escala de resposta de cinco pontos, que varia de "Concordo totalmente" a "Discordo totalmente", também conhecido como escala de Likert [Likert, 1932]. Esta ferramenta é amplamente empregue para avaliar a usabilidade de produtos e serviços, proporcionando resultados confiáveis e permitindo a distinção entre sistemas utilizáveis e não utilizáveis. O questionário que emprega esta técnica é disponibilizado no anexo B.4. É ainda importante mencionar que aos utilizadores foi disponibilizado um Macbook Air de 2019 para efetuarem os testes.

5.4.2 Resultados

Tal como mencionado na secção anterior, cada um dos cinco participante completou quatro tarefas que consistem em cenários distintos de utilização da plataforma. Com o objetivo de facilitar a análise dos resultados alcançados, foram realizados diagnósticos individualizados para cada tarefa - em formato de tabela -, que permitiram avaliar parâmetros como a taxa de sucesso na execução da tarefa, a relação entre o número esperado de cliques e a média real, bem como os tempos decorridos. Estas tabelas forneceram uma visão detalhada e comparativa dos resultados obtidos em cada tarefa, permitindo uma avaliação objetiva e quantitativa da eficiência e desempenho da plataforma. Desta forma, foi possível identificar áreas de melhoria e otimização, com o objetivo de aperfeiçoar a usabilidade e a experiência geral dos utilizadores.

1ª Tarefa

A primeira tarefa consistiu em pesquisar e selecionar três localizações na plataforma. Esta tarefa requer que o utilizador navegue até ao *dashboard*, utilize a barra de pesquisa para encontrar as localizações requisitadas, e use o mapa para as selecionar. Com isto, pretendeu-se verificar se a posição da barra de pesquisa é natural, os controlos do mapa são intuitivos e, no geral, se a interface fornece

¹²Por uma questão de brevidade, apenas apresentamos aqui uma descrição simplificada das tarefas a realizar. As descrições completas das tarefas em questão podem ser consultadas no anexo B.3.

o feedback necessário para que o utilizador reconheça que a localização foi corretamente selecionada e adicionada à lista de localizações, e que esta pode ser alterada posteriormente. Os resultados desta tarefa são expostos na tabela 5.12.

Taxa de sucesso	Nº previsto de cliques	Nº médio de cliques
100%	9	15
Tempo médio (s)	Tempo mínimo (s)	Tempo máximo (s)
51	32	70

Tabela 5.12: Resultados da primeira tarefa do teste de usabilidade

Alguns utilizadores nunca utilizaram um computador cujo sistema operativo é macOs, o que pode causar alguma estranheza inicial. Por exemplo, o Macbook Air 2019 (dispositivo onde foram efetuados todos os testes) exige que se carregue com alguma força no *trackpad* para efetivamente clicar, enquanto que no Windows, um pequeno toque ao de leve com os dedos ativa o clique. Isto revelou-se um entrave grande para alguns utilizadores, ao longo da realização dos vários testes.

2ª Tarefa

A segunda tarefa propôs a filtragem de valores no *dashboard*. Isto requeria que o utilizador intuisse a localização dos filtros e o seu funcionamento. De seguida, foi pedido que o utilizador abrisse a barra inferior - *Location Details* -, para conferir os dados e que escondesse todas as localizações selecionadas, com exceção de uma. Esta tarefa permitiu avaliar a eficiência e facilidade de navegação no *dashboard*. Adicionalmente, possibilitou a avaliar a coesão e intuitividade das ligações lógicas entre os componentes mais importantes do *dashboard*. Os resultados desta tarefa são expostos na tabela 5.13.

Taxa de sucesso	Nº previsto de cliques	Nº médio de cliques
100%	5	12
Tempo médio (s)	Tempo mínimo (s)	Tempo máximo (s)
93	74	130

Tabela 5.13: Resultados da segunda tarefa do teste de usabilidade

A disparidade entre o tempo mínimo e máximo necessário para executar a tarefa, tal como o elevado número de cliques em relação aos realmente necessários, permitiu concluir que a maioria dos participantes tentou diferentes alternativas para alcançar a interação esperada. Esta é uma ocorrência comum, uma vez que foi a primeira vez que estes interagiram com funcionalidades diferentes das que já tinham utilizado em aplicações de meteorologia, por exemplo. Para além disso, a baixa familiaridade de alguns utilizadores com o tipo de interface apresentado (ou com os elementos que a compõem) - seja por fatores como uma idade relativamente avançada ou a menor inclusão informática -, tornou a execução da tarefa mais demorada. Ainda assim, os valores médios dizem-nos que estes resultados são satisfatórios para uma tarefa complexa como a descrita.

3ª Tarefa

A terceira tarefa exigia a utilização da funcionalidade de IDW com base no valor de uma dada variável climática a uma altura fixa, para as localizações selecionadas. Foi proposto ao utilizador que entrasse no modo de desenho do IDW e desenhasse um triângulo¹³ que envolvesse uma qualquer área geométrica à escolha. Esta tarefa foi útil para perceber se os comandos que controlam o IDW são visíveis e localizados num sítio lógico da interface. Para além disso, permite-nos inferir se a maneira como a funcionalidade em si está construída é perceptível para o utilizador à primeira vista. As métricas relativas a esta tarefa estão elicitadas na tabela 5.14.

Taxa de sucesso	Nº previsto de cliques	Nº médio de cliques
100%	8	15
Tempo médio (s)	Tempo mínimo (s)	Tempo máximo (s)
94	65	118

Tabela 5.14: Resultados da terceira tarefa do teste de usabilidade

Os resultados mostram-nos que os utilizadores demoraram algum tempo a encontrar os elementos da interface responsáveis por controlar o desenho da superfície sobre a qual o IDW é renderizado. Para além disso, os utilizadores demonstraram alguma dificuldade em perceber como o próprio modo de desenho funcionava: já com o modo de desenho ativo, alguns sujeitos de teste tentaram clicar e arrastar, quando o gesto necessário era apenas clicar para desenhar cada vértice da superfície. Esta tarefa ajudou-nos a perceber que esta funcionalidade, apesar de funcionar como esperado, tem algumas deficiências a nível de usabilidade.

4ª Tarefa

A 4ª e última tarefa obriga o utilizador a navegar até à página de *Settings*, onde lhe é proposto fazer o upload de um ficheiro CSV na plataforma. Esta tarefa envolve ativar o modo de edição das definições, aceder ao formulário de *upload* e preenchê-lo com os dados requisitados. Este teste permitiu-nos avaliar um dos fluxos mais importantes na utilização desta plataforma no terreno, analisando se o processo de carregamento de informação para a plataforma era rápido, eficaz e intuitivo. Os resultados desta tarefa estão listados na tabela 5.15.

Mais uma vez, a disparidade que observámos na 2ª tarefa (secção ??) está mais uma vez presente - alguns utilizadores utilizaram maneiras mais rápidas de inserir os dados, outros utilizaram formas mais lentas. No entanto, podemos concluir que os resultados são satisfatórios, principalmente para uma tarefa tão relevante no registo e armazenamento da informação das medições feitas no terreno.

¹³A plataforma suporta o desenho de qualquer forma geométrica. No entanto, para efeitos de teste e medição da performance dos utilizadores, é aconselhável fixar essa variável para que o número de cliques seja uma métrica fiável, prevenindo que seja inflacionada pelo facto de um utilizador desenhar uma forma geométrica com mais vértices - e, portanto, exigisse mais cliques.

Taxa de sucesso	Nº previsto de cliques	Nº médio de cliques
100%	15	17
Tempo médio (s)	Tempo mínimo (s)	Tempo máximo (s)
79	61	98

Tabela 5.15: Resultados da quarta tarefa do teste de usabilidade

5.4.3 Análise e Reflexão

Esta secção da validação diz respeito à análise do feedback dos utilizadores, extraído com recurso ao questionário pós-teste, já mencionado acima. Pos uma questão de brevidade, os gráficos correspondentes ao feedback dos sujeitos de teste estão apresentados no anexo B.4, e como tal, todas as referências a figuras nesta secção dizem respeito às imagens do mesmo.

Como já foi mencionado, os utilizadores afirmam que a plataforma é fácil de utilizar (figura B.3), apesar de alguns acharem que necessitariam de um perito para a utilizar (figura B.4). É fácil perceber porquê: algumas das funcionalidades são direcionadas a peritos na área da meteorologia - ou pelo menos, da área científica -, e portanto poderão ser algo confusas para pessoas que não tenham formação na matéria (por exemplo, o desenho do IDW). É importante salientar que uma minoria de utilizadores não pertencem a área onde seja comum lidar com interfaces do tipo da apresentada.

É de notar que, apesar de alguma possível estranheza ou confusão inicial durante o primeiro contacto com a plataforma (figura B.10), todos os utilizadores que foram sujeitos a teste pensam que outros utilizadores irão rapidamente aprender a utilizar a plataforma (figura B.7). De forma geral, o feedback dos utilizadores mostra que a curva de aprendizagem para começar a utilizar os básicos da plataforma não é muito íngreme.

No entanto, em conjunto com as métricas extraídas na segunda e terceira tarefas dos testes de utilizador (secções 5.4.2 e 5.4.2), os resultados mostram que algumas funcionalidades não são muito explícitas ou intuitivas - apesar de, em média, se afirmarem confiantes (figura B.9), os utilizadores demoraram mais tempo do que o expectável para tarefas com um número tão reduzido de cliques previstos. No caso específico do IDW e os botões que controlam a sua renderização, é claro que é necessária uma reformulação ou uma alteração da abordagem que estamos a utilizar para apresentar essa funcionalidade na interface atual. Uma possível solução poderá passar por um tutorial do funcionamento do *dashboard*, integrado na plataforma em si, que poderia ser mostrado ao utilizador aquando da sua primeira utilização.

De modo global, estes testes permitiram-nos avaliar adequadamente a intuitividade e facilidade de uso da nossa plataforma, até para utilizadores em áreas pouco relacionadas com a climatologia e a meteorologia. O conjunto de métricas escolhidas iluminaram algumas dúvidas que nos foram aparecendo durante o desenvolvimento da plataforma, e confirmaram algumas suspeitas já formuladas. O processo de teste foi útil para descobrir algumas deficiências da plataforma e per-

mitiu que melhor entendêssemos as suas forças. Tendo em conta o feedback dos utilizadores testados, podemos afirmar que a plataforma desenvolvida constitui uma boa base por onde partir para um produto mais elaborado, caso seja esse o objetivo.

Capítulo 6

Conclusão

A dissertação apresentada representa o culminar de um trabalho aprofundado sobre a importância e os benefícios das plataformas de visualização na extração e aquisição de conhecimento. Ao longo deste processo, foram abordados diversos tópicos relevantes, desde símbolos cartográficos e recursos de visualização básicos até técnicas de interação complexas e plataformas de apoio à decisão, apoiadas na visualização analítica. A análise detalhada do estado da arte revelou que as visualizações, elementares ou complexas, desempenham um papel vital no desenvolvimento de soluções inovadoras ou simplesmente no entendimento e percepção que se tem de um certo problema. Estas surgem como uma ferramenta fundamental para entidades da área científica garantirem a consistência e eficiência nos seus projetos de investigação. Os casos de estudo apresentados ao longo desta investigação evidenciam os impactos positivos alcançados com a implementação de plataformas de visualização, incluindo uma maior disseminação do conhecimento obtido, tanto entre indivíduos do meio científico, como para o público geral.

No âmbito do projeto desenvolvido, foi criada uma plataforma abrangente e completa. A disponibilidade de uma ferramenta deste tipo permite que os utilizadores acedam a informação fidedigna em qualquer momento do processo de recolha ou análise dos dados. Esta vantagem possibilita, assim, agilizar a investigação e o processo de interpretação da informação adquirida. Isto é particularmente relevante, uma vez que não existe uma oferta generalizada deste tipo de capacidades em projetos públicos atuais. Ao considerar estes factores, os investigadores podem compreender e analisar os seus próprios dados e a tomar decisões informadas, sem a intervenção de um perito do domínio da visualização ou processamento de dados. Acreditamos que foi desenvolvida uma plataforma que torna os seus utilizadores mais independentes e mais conhecedores no domínio da climatologia.

Em suma, esta dissertação contribui para a compreensão aprofundada das plataformas de visualização como uma ferramenta essencial no contexto científico, destacando a sua relevância na melhoria dos processos de investigação e recolha de dados e na obtenção de resultados mais eficientes e consistentes. O trabalho desenvolvido revela-se como um ponto de partida para futuras investigações e implementações de plataformas semelhantes em outras organizações ou domí-

nios, visando potenciar o sucesso e a qualidade dos projetos de visualização.

6.1 Trabalho Realizado

No decorrer desta dissertação, foi concluída uma primeira versão da plataforma proposta, pronta para ser utilizada por indivíduos ou grupos da área da climatologia. A ferramenta foi desenvolvida como um agregado de várias técnicas de visualização que potenciam a investigação e a exploração dos dados adquiridos.

Além disso, foram realizados testes de usabilidade que possibilitaram a validação formal das escolhas de implementação feitas durante a conceção e desenvolvimento da plataforma, assim como a recolha de feedback por parte de alguns utilizadores. A análise dos resultados destes testes mostraram-se encorajadores, ilustrando a necessidade da existência de uma plataforma deste tipo.

O trabalho desenvolvido permitiu verificar a aplicabilidade e desempenho de uma solução deste tipo em situações reais, fornecendo informações valiosas para a sua contínua melhoria.

6.2 Trabalho Futuro

No que diz respeito ao trabalho futuro, existem várias oportunidades para expandir e aprofundar ainda mais a plataforma desenvolvida. Primeiramente, é essencial garantir a sua evolução contínua, adaptando-o às necessidades em constante mudança dos utilizadores. Isto envolve reavaliar regularmente o conjunto de funcionalidades existente e adicionar novas, caso haja necessidade.

Nesta ótica, foram identificadas algumas possíveis melhorias e funcionalidades adicionais, que, não tendo sido implementadas, foram pelo menos, conceptualizadas posteriormente à implementação do MVP:

- **Configuração de unidades** - deve ser permitido ao utilizador identificar as unidades dos dados que carrega para a plataforma. Não é raro acontecer que as unidades utilizadas numa dada medição não são as unidades mandadas pelo Sistema Internacional de unidades (p.e. km/h vs m/s). Assim que haja a indicação da unidade, o sistema pode fazer a conversão automática da unidade SI padrão para a unidade inserida pelo utilizador.
- **Auto-play em modo range** - Permitir que o utilizador itere sobre os valores contidos num dado intervalo selecionado, sem que seja preciso a sua interação. Esta técnica possibilitaria ao utilizador visualizar a progressão de uma certa variável climática durante um dado intervalo de tempo ou de altitude. Seria possível fazer *play* e *pause*, ao estilo de um *media player*.
- **Upload de ficheiros em massa** - Carregamento de vários ficheiros de uma vez só. Esta funcionalidade facilitaria

- *Forecasting* - Integração de modelos de previsão de séries temporais para completar valores em falta. Por exemplo, se um utilizador quiser analisar a humidade a partir de uma altitude de 150 metros, mas apenas tem ao seu dispôr medições até uma altitude de 100 metros, poderá ser possível inserir artificialmente esses valores de forma relativamente fidedigna, possibilitando uma análise minimamente informada, sem exigir mais medições.

Ao considerar estas oportunidades de trabalho futuro, qualquer entidade que utilize a plataforma desenvolvida irá maximizar os seus benefícios e continuar a impulsionar o sucesso das suas investigações. A manutenção contínua do sistema, a avaliação e feedback regulares, bem como a pesquisa e acompanhamento das tendências do setor são fundamentais para garantir a eficácia da plataforma no contexto científico.

Referências

- Bach, B., Freeman, E., Abdul-Rahman, A., Turkay, C., Khan, S., Fan, Y., and Chen, M. (2022). Dashboard design patterns. *arXiv preprint arXiv:2205.00757*.
- Barreto, M. (1993). The dot map as an epidemiological tool: A case study of schistosoma mansoni infection in an urban setting. *International journal of epidemiology*, 22:731–41.
- Bartier, P. M. and Keller, C. P. (1996). Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (idw). *Computers & Geosciences*, 22(7):795–799.
- Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. (2002). Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 259–266.
- Bertin, J. (1983). *Semiology of graphics*. University of Wisconsin press.
- Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE software*, 8(1):32–41.
- Bostock, M., Ogievetsky, V., and Heer, J. (2011). D³ data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309.
- Brehmer, M., Sedlmair, M., Ingram, S., and Munzner, T. (2014). Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pages 1–8.
- Brewer, C. A. and Pickle, L. (2002). Evaluation of methods for classifying epidemiological data on choropleth maps in series. *Annals of the Association of American Geographers*, 92(4):662–681.
- BUL (2020). Bul - banda ultra larga italia.
- Card, M. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.
- Centre, B. F. S. (2022). Choropleth maps.
- Chua, A. and Moere, A. V. (2017). Binsq: visualizing geographic dot density patterns with gridded maps. *Cartography and Geographic Information Science*, 44(5):390–409.

- Cockburn, A., Karlson, A., and Bederson, B. B. (2009). A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):1–31.
- Cook, K. A. and Thomas, J. J. (2005). Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States).
- Covid19India (2021). Coronavirus in india: Latest map and case count.
- Dent, B. D. (1999). *Cartography: Thematic map design*. Ingram.
- Dey, P. K., Kinch, J., and Ogunlana, S. O. (2007). Managing risk in software development projects: a case study. *Industrial Management & Data Systems*, 107(2):284–303.
- Dupin, C. (1826). Table of arts and crafts and fine arts: present to serve to propagate the institution of the courses of géométry and méchanics applied to the arts, in the cities of france. 2.
- Fedosejev, A. (2015). *React. js essentials*. Packt Publishing Ltd.
- Friendly, M. (2002). Visions and re-visions of charles joseph minard. *Journal of Educational and Behavioral Statistics*, 27:31–51.
- GLEAM (2022). *GleamViz UI*. GLEAM.
- Google (2005). Google maps.
- Haughey, D. (2011). Moscow method. *Project Smart*, page 2011.
- Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., and Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In *Information visualization*, pages 154–175. Springer.
- Keim, D. and Zhang, L. (2011). Solving problems with visual analytics: Challenges and applications. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, pages 1–4.
- Keim, D. A. (2002). Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8.
- Kimerling, A. J. (2009). Dotting the dot map, revisited. *Cartography and Geographic Information Science*, 36(2):165–182.
- Kloeckl, K., Anwar, K., Bhatnagar, R., Cruz, P., Nagel, T., and Senn, O. (2012). Visual explorations of urban mobility.
- Koch, W. G. (2000). Jaques bertin’s theory of graphics and its development and influence on multimedia cartography. *Information Design Journal*, 10(1):37–43.
- Kotonya, G. and Sommerville, I. (1998). *Requirements engineering: processes and techniques*. Wiley Publishing.

- Kraak, M.-J. (2003). Geovisualization illustrated. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57(5):390–399. Challenges in Geospatial Analysis and Visualization.
- Kyriakidis, A., Maniatis, K., and You, E. (2016). *The majesty of Vue. js*. Packt Publishing.
- Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.
- Loomis, E. (1845). On two storms which were experienced throughout the united states, in the month of february, 1842. *Transactions of the American Philosophical Society*, 9(2):161–184.
- MacEachren, A. M. (1979). The evolution of thematic cartography: A research methodology and historical review. *The Canadian Cartographer*, 16(1):17–33.
- MDN (2023). Event reference | mdn.
- Meirelles, I. (2013). *Design for information: an introduction to the histories, theories, and best practices behind effective information visualizations*. Rockport publishers.
- Minard, C.-J. (1869). Tableaux graphiques et cartes figuratives de m. minard, 1845-1869.
- Munzner, T. (2009). A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6):921–928.
- Munzner, T. (2014). *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press.
- Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.
- Parallel (2020). London maps 3d: Laei no2.
- Parent-Duchatelet, A. J. B. (1836). *Prostitution in the city of Paris: considered in relation to public health, morality and administration: work supported by statistical documents drawn from the archives of the Préfecture of police*, volume 2. Hauman, Cattoir.
- Peuquet, D. J. (2002). *Representations of space and time*. Guilford Press.
- Phan, D., Xiao, L., Yeh, R., and Hanrahan, P. (2005). Flow map layout. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 219–224. IEEE.
- Pincetl, S., Federico, F., Gustafson, H., Strug, L., Fournier, E., and Cudd, R. (2020). About the ucla energy atlas project.
- Polisciuc, E. (2021). *Thematic Cartography for Adaptive Visualization Systems*. PhD thesis, 00500:: Universidade de Coimbra.
- Pons, A., Rullán, O., and Murray, I. (2014). Tourism capitalism and island urbanization: tourist accommodation diffusion in the balearics, 1936-2010. *Island Studies Journal*, 9(2):239–258.

-
- Robinson, A. H. (1960). Elements of cartography. *Soil Science*, 90(2):147.
- Robinson, A. H. and Wallis, H. M. (1967). Humboldt's map of isothermal lines: A milestone in thematic cartography. *The Cartographic Journal*, 4(2):119–123.
- Roth, R. E. (2017). Visual variables. *International encyclopedia of geography: People, the earth, environment and technology*, pages 1–11.
- Russo, A. J. (2003). Amazon quicksight.
- Sandvik, B. (2008). Using kml for thematic mapping. *Institute of Geography School of GeoSciences. Edinburgh, University of Edinburgh. MSc in Geographical Information Science*, 22.
- Slocum, T. A., McMaster, R. M., Kessler, F. C., Howard, H. H., and Mc Master, R. B. (2008). Thematic cartography and geographic visualization.
- Soetens, L., Hahné, S., and Wallinga, J. (2017). Dot map cartograms for detection of infectious disease outbreaks: an application to q fever, the netherlands and pertussis, germany. *Eurosurveillance*, 22(26):30562.
- Taylor, P. (2022). Total data volume worldwide 2010-2025.
- Taylor, P. (2023). Data growth worldwide 2010-2025.
- Tennekes, M. (2018). Aggregation of dots. *Methods for Big Data in Official Statistics*, 8.
- vis.gl (2017). React-map-gl | clusters example - github pages.
- Woodbridge, W. C. (1823). Isothermal chart, or view of climates & production, drawn from the accounts of humboldt & others. *Woodbridge's School Atlas to Accompany Woodbridge's Rudiments of Geography: Atlas on a New Plan, exhibiting the Prevailing Religions, Forms of Government, Degrees of Civilization, and the Comparative Size of Towns, Rivers and Mountains. Oliver D. Cooke & Co., Hartford, Conn.*
- WorldMapper (2020). Us presidential election 2020.
- Xiaosong, L., Shushi, L., Wenjun, C., and Songjiang, F. (2009). The application of risk matrix to software project risk management. In *2009 International Forum on Information Technology and Applications*, volume 2, pages 480–483. IEEE.
- Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231.
- You, E. (2023). Frequently asked questions | vue.js.

Anexos

Apêndice A

Requisitos Funcionais

ID	Descrição	Prioridade
RF001	Iniciar sessão	M
RF002	Terminar sessão	M
RF003	Criar conta	M

Tabela A.1: Requisitos Funcionais de Autenticação

ID	Descrição	Prioridade
RF004	Escolher o sistema de unidades preferido	M
RF005	Alterar o nome de perfil	M
RF006	Alterar a visibilidade de variáveis no <i>dashboard</i>	M
RF007	Adicionar uma variável nova à plataforma	M
RF008	Adicionar uma localização nova à plataforma	M
RF009	Adicionar uma medição nova à plataforma	M
RF010	Adicionar novos valores através de um ficheiro CSV	M
RF011	Remover uma variável da plataforma	C
RF012	Remover uma localização da plataforma	C
RF013	Remover uma medição da plataforma	C

Tabela A.2: Requisitos Funcionais das definições de utilizador

ID	Descrição	Prioridade
RF014	Visualizar num mapa as localizações disponíveis	M
RF015	Selecionar num mapa as localizações disponíveis	M
RF016	Remover uma localização das localizações selecionadas	M
RF017	Alterar a visibilidade de uma localização selecionada	M
RF018	Pesquisar pelo nome as localizações disponíveis	M
RF019	Visualizar interpolação de valores das localizações selecionadas no mapa	M
RF020	Filtrar os valores medidos por altitude	M
RF021	Filtrar os valores medidos por momento no tempo	M
RF022	Filtrar os valores medidos por intervalos de valores	M
RF023	Filtrar os valores medidos por valores específicos	M
RF024	Visualizar a distribuição dos valores medidos	M
RF025	Visualizar as médias dos valores medidos em função da altitude	M
RF026	Visualizar os valores medidos em função da altitude	M
RF027	Alternar a variável climática visualizada em detalhe	M

Tabela A.3: Requisitos Funcionais de Visualização

Apêndice B

Testes de Usabilidade

B.1 Documento de Consentimento Informado

Título do Estudo: Avaliação da eficácia da utilização da plataforma para visualização de dados climáticos.

Instituição: Departamento de Engenharia Informática.

Os convidados para este estudo são profissionais da área de tecnologia, o que lhes confere um nível de conforto no uso de dispositivos móveis e interfaces correspondentes, permitindo-lhes adotar uma perspetiva crítica neste estudo. Os participantes deste estudo irão realizar quatro tarefas que correspondem aos fluxos principais da plataforma. Inicialmente, serão recolhidos dados pessoais e demográficos, seguidos pela conclusão das tarefas propostas utilizando a plataforma. Por fim, será solicitado que preencham um formulário que permitirá avaliar a usabilidade da plataforma. A participação no estudo é totalmente voluntária, e os participantes podem optar por sair do estudo a qualquer momento, informando verbalmente o investigador. O investigador também pode excluir um participante do estudo caso este não cumpra as diretrizes estabelecidas. Todas as informações recolhidas serão mantidas confidenciais, e o nome dos participantes será substituído por um número de código ao longo de todo o estudo. Declaro que li o formulário e aceito fazer parte deste estudo de livre vontade.

Nome completo: _____

Assinatura: _____

B.2 Questionário Pré-Teste

Tal como mencionado no guião, cada participante deverá começar por responder a um breve formulário com questões de carácter pessoal que procuram conhecer melhor o perfil da pessoa e a sua envolvência no meio tecnológico. O questionário partilhado com os participantes poderá ser consultado [aqui](#).

B.3 Tarefas

1. Pesquisar e selecionar três localizações

1. Esta tarefa inicia-se na página *Dashboard*.
2. Deverá selecionar a barra de pesquisa, escrever o nome "Linhares 1" e selecionar o resultado que aparece.
3. Selecionar a localização que resultou da pesquisa.
4. Repetir os dois passos anteriores com os nomes "Rocha Nova" e "Casa do Sal".

As respostas dos sujeitos de teste estão representados na seguinte tabela (dividida em várias para facilitar a compreensão).

Idade	Género	Escolaridade	Área de formação	Profissão
23	M	Ensino Superior	Ciência de Dados	Cientista de Dados
59	M	Ensino Superior	Eng. Eletrotécnica	Eng. Eletrotécnico
25	F	Ensino Sup.	Gestão	Bailarina
30	F	Ensino Sup.	Medicina	Médica
41	M	Ensino Sup.	Eng. Informática	Eng. Informático

Horas online/dia	Websites/ Aplicações
7 - 9	Youtube, Messenger, WhatsApp, Slack.
10 - 12	Outlook, Microsoft Teams, Excel, Gmail, Sapo.pt.
4 - 6	Instagram, TikTok, Youtube.
4 - 6	Youtube, HBO para entretenimento.
7 - 9	Slack, WhatsApp. Netflix, Spotify.

Dispositivos	OS	Plataforma utilizada com funcionalidades semelhantes
Computador, Telemóvel	macOS	A app de metereologia pré-definida, no Iphone.
Computador, Telemóvel	Windows	A app de metereologia da Huawei, no telemóvel.
Telemóvel	Windows	A app de metereologia da Huawei, no telemóvel.
Telemóvel	Windows	A app de metereologia do telemóvel.
Computador, Telemóvel	macOS	A weather app do Iphone.

2. Filtragem

1. Esta tarefa inicia-se na página *Dashboard*.
2. Abrir o filtro Temperature.
3. Selecionar o intervalo de temperaturas 3 a 13.

4. Abrir o filtro Time.
5. Selecionar o intervalo de datas 1/Janeiro/2019 00:00 à data de hoje e a hora atual.
6. Abrir o filtro Altitude.
7. Selecionar o intervalo de altitudes 50 a 180, arrastando os indicadores.
8. Abrir a barra Location Details.
9. Esconder todas as localizações, exceto a localização Rocha Nova.

3. Utilização do IDW

1. Esta tarefa tarefa inicia-se na página *Dashboard*.
2. Abrir o filtro Altitude.
3. Mudar o seu modo para "value".
4. Arrastar o indicador do gráfico até à altitude 150.
5. Selecionar o modo de desenho.
6. Desenhar uma figura geométrica que englobe um determinada área geográfica.

4. Upload de um ficheiro CSV

1. Esta tarefa tarefa inicia-se na página *Settings*.
2. Ativar o modo de edição.
3. Selecionar o botão de adicionar dados.
4. Digitar o nome "Linhares 1" e prosseguir para o próximo passo.
5. Em Start Timestamp, colocar a data 18/08/2023 e as horas 12h00min.
6. No End Timestamp, colocar a mesma data mas as horas 12h05min.
7. No campo Variable, selecionar Temperature, e prosseguir para o próximo passo.
8. Selecionar o campo de carregamento de um ficheiro CSV.
9. Selecionar o ficheiro "example-data-7".
10. Submeter o formulário.

B.4 Questionário Pós-Teste

Após a realização de todas as tarefas, os participantes responderam a outro formulário que, tal como mencionado na secção , permitiu avaliar a usabilidade da plataforma com base na *System Usability Scale (SUS)*. O questionário partilhado com os participantes poderá ser consultado aqui. Tal como é mencionado na secção Validação, a escala varia de 1 a 5, em que 1 corresponde a "Discordo Completamente", e 5 corresponde a "Concordo Completamente".

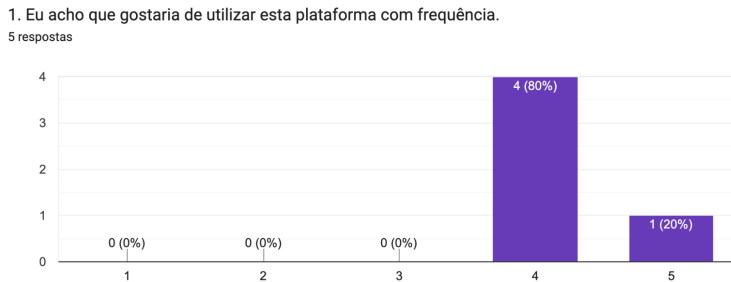


Figura B.1: Resultados da primeira pergunta do questionário pós-teste.

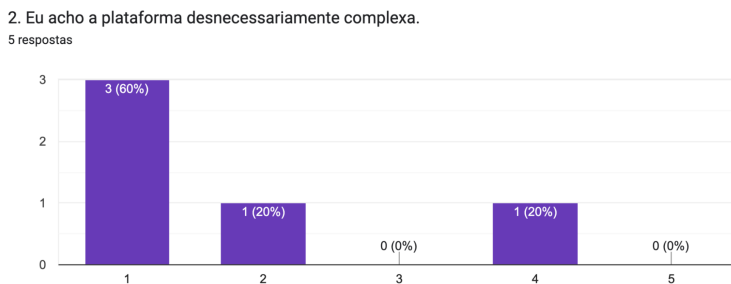


Figura B.2: Resultados da segunda pergunta do questionário pós-teste.

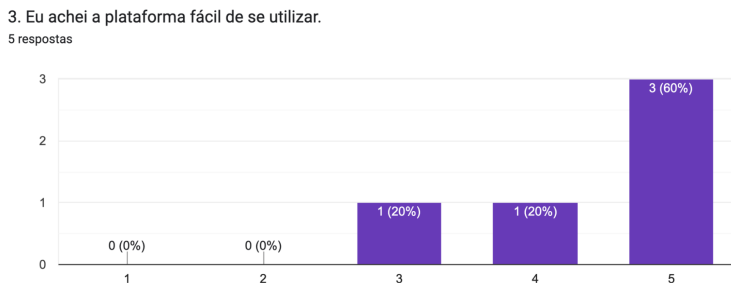


Figura B.3: Resultados da terceira pergunta do questionário pós-teste.

4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para utilizar a plataforma.

5 respostas

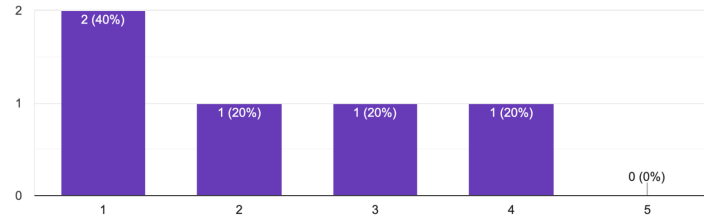


Figura B.4: Resultados da quarta pergunta do questionário pós-teste.

5. Eu acho que as várias funções da plataforma estão muito bem integradas.

5 respostas

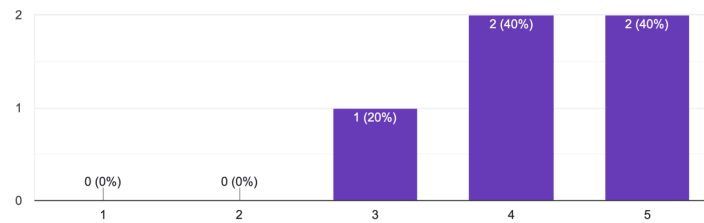


Figura B.5: Resultados da quinta pergunta do questionário pós-teste.

6. Eu acho que a plataforma apresenta muitas inconsistências.

5 respostas

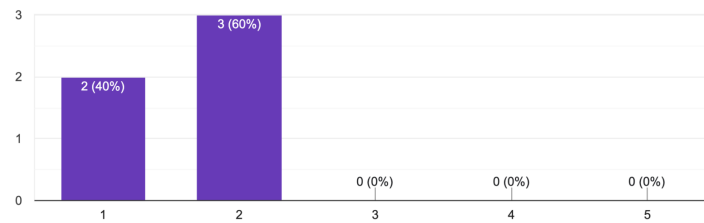


Figura B.6: Resultados da sexta pergunta do questionário pós-teste.

7. Eu penso que as pessoas irão aprender rapidamente a utilizar esta plataforma.

5 respostas

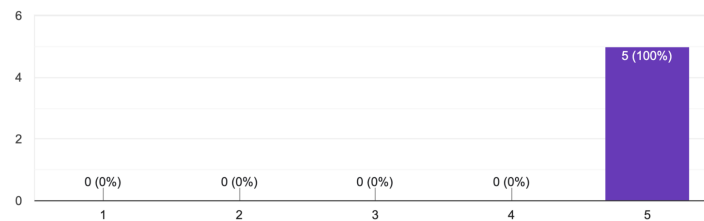


Figura B.7: Resultados da sétima pergunta do questionário pós-teste.

8. Eu achei a plataforma confusa de se utilizar.
5 respostas

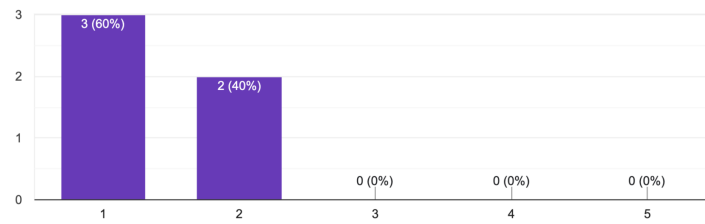


Figura B.8: Resultados da oitava pergunta do questionário pós-teste.

9. Eu senti-me confiante ao utilizar a plataforma.
5 respostas

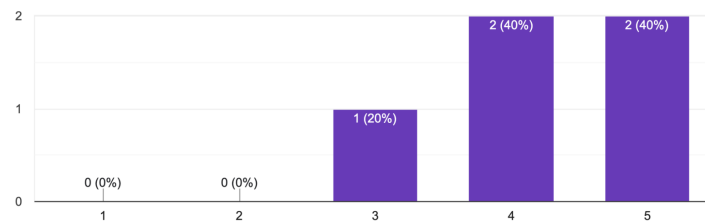


Figura B.9: Resultados da nona pergunta do questionário pós-teste.

10. Foi necessário aprender várias coisas novas antes de conseguir utilizar a plataforma.
5 respostas

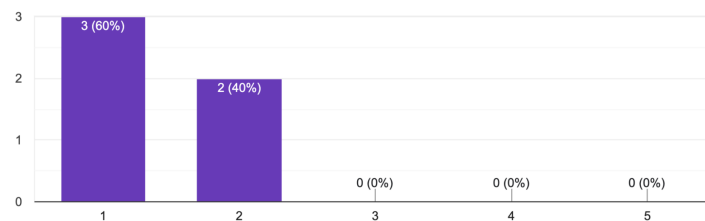


Figura B.10: Resultados da décima pergunta do questionário pós-teste.