

**UNIVERSIDADE ABERTA**



**Desenvolvimento de aplicativo web para visualização de  
informação hidrológica**

Bruno Medeiros Moreira

1900421

Mestrado em Engenharia Informática e Tecnologia Web

2024



**UNIVERSIDADE ABERTA**



**Desenvolvimento de aplicativo web para visualização de  
informação hidrológica**

Bruno Medeiros Moreira

1900421

Mestrado em Engenharia Informática e Tecnologia Web

Dissertação orientada por:

Prof. Doutor Ricardo José Vieira Baptista, UAB

Prof. Doutor Otto Corrêa Rotunno Filho, COPPE/UFRJ

2024

# AGRADECIMENTOS

Agradeço aos meu orientador, Ricardo José Vieira Baptista (UAB/UP), em sua especialidade, foi fundamental para a conclusão deste trabalho. Mais do que essa contribuição, foi, também, além, dando todo o suporte, amizade e oportunidades. Não tenho palavras para agradecer todo o apoio durante toda essa jornada.

Ao meu também orientador Otto Corrêa Rotunno Filho (COPPE/UFRJ), por todos os ensinamentos teóricos e práticos que foram fundamentais para meu crescimento profissional e acadêmico, e, em especial, pela oportunidade e por todo o suporte oferecido para condução deste trabalho.

Aos engenheiros, meteorologistas, professores que contribuíram com a disponibilidade de tempo para as entrevistas e feedback, durante a realização da dissertação.

A meu irmão, Daniel Medeiros Moreira, grande amigo e incentivador da minha pesquisa. Sem ele, jamais teria êxito neste trabalho e, por essa razão, dedico esta dissertação a ele, por toda a sua generosidade e bondade que me possibilitaram chegar até aqui e realizar diversas conquistas.

Por fim, os meus mais profundos agradecimentos a Deus, à minha mãe Sonia Moreira, e ao meu pai Flávio Machado Moreira (*in memoriam*), que olham por mim onde quer que estejam, com apoio incondicional ao longo de toda a minha vida, sendo fundamentais para a minha formação como pessoa.



## DECLARAÇÃO DE INTEGRIDADE STATEMENT OF INTEGRITY

Declaro ter atuado com integridade na elaboração da presente dissertação/tese. Confirmando que em todo o trabalho conducente à sua elaboração não recorri à prática de plágio ou a qualquer outra forma de falsificação de resultados.

Mais declaro que tomei conhecimento integral do Regulamento Disciplinar da Universidade Aberta, publicado no Diário da República, 2.<sup>a</sup> série, n.º 215, de 6 de novembro de 2013.

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged Disciplinary Regulations of the Universidade Aberta (regulation published in the official journal Diário da República, 2.<sup>a</sup> série, N.º 215, de 6 de novembro de 2013).

Universidade Aberta, 24 de Janeiro de 2024

Nome completo/Full name: Bruno Medeiros Moreira

Assinatura/Signature:

---

manuscrita ou digital / handwritten or digital

## Resumo

Os dados geocientíficos ambientais gerados, em especial de natureza hidrológica, são normalmente em formatos complexos e não podem ser entendidos e visualizados facilmente sem um pré-processamento, demandando ferramentas específicas da área de sistemas de informações geográficas, que requerem um conhecimento especializado. Diante desse cenário, os aplicativos web tornam-se um excelente meio para torná-los acessíveis com vistas a atender públicos distintos, quer sejam técnicos ou leigos. No entanto, a experiência necessária para desenvolver tais aplicativos enfrenta alguns desafios em potencial. Em linhas gerais, esses obstáculos podem ser caracterizados pelos seguintes elementos que os desenvolvedores precisam superar: (i) acessibilidade e tratamento de dados, (ii) identificação, seleção e configuração dos softwares requeridos para atender as resoluções espaciais e temporais; (iii) avaliação e configuração da infraestrutura computacional necessária para disponibilizar informação hidrológica; (iv) configuração e programação entre os diferentes arcabouços para o desenvolvimento da interface de utilização (frameworks); (v) construção de gráficos e estatísticas significativas para análise dos dados hidrológicos processados. Nesse sentido, o trabalho propõe-se a investigar ferramentas de código livre e desenvolver um aplicativo web responsivo para acessar a informação hidrológica disponibilizada por missões de satélites e por estações pluviométricas in situ, que, em função dos diversos campos espaço-temporais gerados, requer processamento para visualização. O estudo de caso foi desenvolvido para dados hidrológicos do Brasil gerenciados pela Agência Nacional de Águas e Saneamento Básico (ANA). Foram pesquisadas diversas tecnologias e ferramentas de código aberto para sistemas de informações geográficas e desenvolvimento web em geral, sendo as principais ferramentas selecionadas discriminadas na sequência: Nginx como servidor web com NodeJs, STRAPI como API REST para gerenciamento de conteúdo dos dados hidrológicos, MapLibre como biblioteca que usa WebGL para renderizar mapas interativos, as bibliotecas GDAL/GMT para processamento dos dados hidrológicos e React para produzir a interface de utilizador. Como resultado, o aplicativo permitiu visualizar informação geocientífica de mais de 1500 pontos de interesse pluviométrico no Brasil em forma de gráficos para facilitar a interpretação das séries históricas de chuva.

Palavras-chave: Aplicativo Web, Hidrologia, Sistema de Informações Geográficas.

## Abstract

The environmental geoscientific data generated, especially of a hydrological nature, are normally in complex formats and cannot be understood and visualized easily without pre-processing, requiring specific tools in the area of geographic information systems, which require specialized knowledge. Given this scenario, web applications become an excellent means of making them accessible with a view to serving different audiences, whether technical or lay. However, the expertise required to develop such applications faces some potential challenges. In general terms, these obstacles can be characterized by the following elements that developers need to overcome: (i) accessibility and data processing, (ii) identification, selection and configuration of software required to meet spatial and temporal resolutions; (iii) assessment and configuration of the computational infrastructure necessary to provide hydrological information; (iv) configuration and programming between different frameworks for the development of the user interface (frameworks); (v) construction of significant graphs and statistics for analysis of processed hydrological data. In this sense, the work proposes to investigate free code tools and develop a responsive web application to access hydrological information made available by satellite missions and in situ rain gauge stations, which, due to the different spatio-temporal fields generated, requires processing for visualization. The case study was developed for Brazilian hydrological data managed by the National Water and Basic Sanitation Agency (ANA). Several open-source technologies and tools for geographic information systems and web development in general were researched, with the main selected tools being broken down as follows: Nginx as a web server with NodeJs, STRAPI as a REST API for content management of hydrological data, MapLibre as library that uses WebGL to render interactive maps, the GDAL/GMT libraries for processing hydrological data and React to produce the user interface. As a result, the application made it possible to visualize geoscientific information from more than 1500 points of rainfall interest in Brazil in the form of graphs to facilitate the interpretation of historical rainfall series.

Keywords: Web Application, Hydrology, Geographical Information System



# ÍNDICE

1. Introdução.....	1
1.1 Contexto e motivação .....	1
1.2 Problema de investigação .....	4
1.3 Objetivos do Projeto .....	8
1.4 Metodologia de Investigação .....	8
1.5 Resultados Esperados .....	9
1.6 Organização do Documento.....	10
2. Bases Teóricas e Revisão Bibliográfica .....	12
2.1 Coleta, acesso e observação de dados de precipitação .....	13
2.1.1 Monitoramentos de precipitação em superfície .....	13
2.1.2 Uso de satélite para obter dados precipitação .....	16
2.1.3 Processamento de imagens de precipitação por Satélite.....	19
2.1.4 Coleta de dados de precipitação in situ no Brasil .....	22
2.2 Repositórios de dados hidrológicos .....	25
2.3 Padrões de Sistemas de Informação Geográfica.....	27
2.3.1 Web Map Service (WMS).....	28
2.3.2 Feature Service (WFS).....	28
2.3.3 Web Coverage Service (WCS).....	28
2.4 Tecnologias de software gratuita e de código aberto para área geoespacial.....	29
2.4.1 MapServer .....	30
2.4.2 GeoServer.....	30
2.4.3 OpenStreetMap, TileServer, OpenMapTiles.....	31
2.4.4 Frameworks para camada de interface de usuários para Mapas .....	32
2.5 Tecnologias de desenvolvimento na web .....	36
2.5.1 Formas de desenvolvimento para Web .....	37
2.5.2 Frameworks de Desenvolvimento em Javascript.....	39
2.5.3 Responsive Web Design .....	40
2.6 Experiência de Usuário (UX) e Design UX.....	43
2.6.1 Padrões de Interface de Usuário.....	45
3. Metodologia.....	48
3.1 Design Research Science .....	49
3.1.1 Ciclos de engenharia .....	51
3.1.2 As Partes Interessadas (Stakeholders) e o Contexto Social .....	51
3.2 Seleção de técnicas, abordagens e processos de Engenharia de Software.....	53
3.2.1 Requisitos do Software .....	56
3.2.2 Técnicas de Elicitação de Requisitos .....	56

3.2.3	Identificação das fontes dos requisitos.....	57
3.2.4	Modelagem do Sistema e dos Requerimentos .....	57
3.2.5	Projeto de Arquitetura .....	57
3.2.6	Implementação de Componentes .....	58
3.2.7	Teste de Software.....	58
4.	Requisitos e Projeto de Arquitetura.....	59
4.1	Entrevistas.....	60
4.2	Elicitação de Requisitos.....	66
4.2.1	Requisitos Funcionais .....	66
4.2.2	Requisitos não funcionais .....	67
4.3	Escolha de Componentes .....	67
4.3.1	Seleção de Tecnologias Backend.....	69
4.3.2	Seleção de Tecnologias Frontend.....	71
4.4	Projeto de Arquitetura.....	72
5.	Lado do Servidor (Backend) e Tratamento de Dados .....	75
5.1	Aquisição e processamento dados insitu .....	76
5.2	Aquisição e processamento dados de satélite .....	77
5.3	API REST – STRAPI .....	79
6.	Interface de Usuário e Desenvolvimento do Protótipo.....	81
6.1	UX (Experiência de Usuário) para interface de usuário .....	82
6.2	Implementação Física .....	87
6.2.1	Implementação em ambiente de produção .....	88
6.3	Implementação de Design Responsivo .....	90
7.	Testes e Resultados.....	93
7.1	Testes Manuais e Automatizados .....	94
7.2	Avaliação de Experiência de Usuário .....	96
8.	Conclusões e Recomendações.....	101
	Referências bibliográficas .....	106
	Anexo I - Questionário de Experiência de Usuário.....	109

## ÍNDICE DE FIGURAS

Figura 2.1	Equipamentos para medição de precipitação	15
Figura 2.2	Conceituação da observação de precipitação da missão GPM	19
Figura 2.3	Imagem digital de sensoriamento remoto em camadas	20
Figura 2.4	Fluxograma de calibração/validação de produtos de precipitação	21
Figura 2.5	Fluxo de dados da rede hidrometeorológico nacional de referência	23
Figura 2.6	Densidade de estações pluv. (esq.) e fluv. (dir.) por região hidrográfica	24
Figura 2.7	Como OSM é transformado em bloco de vetores.	32
Figura 2.8	Relação entre fontes de dados, tilesets e estilos	33
Figura 2.9	Openlayers, índice de verde da vegetação	34
Figura 2.10	Camada raster Web Map Service (WMS)	35
Figura 2.11	Linguagens mais utilizados pelos desenvolvedores	36
Figura 2.12	Diferença entre biblioteca e framework	37
Figura 2.13	Navegadores por tipo de dispositivo	41
Figura 2.14	Ciclo de vida de atividades de design UX	44
Figura 2.15	Componente Card do framework Bootstrap.	46
Figura 3.1	Artefatos e Contexto	50
Figura 3.2	Ciclos de engenharia	51
Figura 3.3	Contexto do projeto de ciência do design	52
Figura 3.4	Paradigma de prototipagem	55
Figura 4.1	Diagrama de caso de uso da aplicação web MapHidro	67
Figura 4.2	Arquitetura de alto nível do aplicativo	73
Figura 4.3	Diagrama de arquitetura com componentes especificados	73
Figura 5.1	Repositório e extrato de arquivo CVS do repositório da ANA	76
Figura 5.2	Extrato de código para processar dados de precipitação	77
Figura 5.3	Diagrama de sequência para geração do mapa de calor de precipitação	78
Figura 5.4	Extrato de código processar precipitação de satélite	78
Figura 5.5	Estações pluviométricas do MapHidro no STRAPI	79
Figura 5.6	Excerto de código para inserção de precipitação na API	80
Figura 6.1	Personas criadas com entrevistas	83
Figura 6.2	Fluxo de tarefas do sistema	84
Figura 6.3	Mapa de navegação	85
Figura 6.4	Wireframe Gráficos de Média de Chuva	86
Figura 6.5	Wireframe Resultados de Busca	86
Figura 6.6	Wireframes para dispositivo móvel	87
Figura 6.7	Arquitetura de implantação da camada de suporte	88
Figura 6.8	Github do projeto	89
Figura 6.9	Excerto código componente Map.tsx	90
Figura 6.10	Protótipo da página inicial	91
Figura 6.11	Janela Pop-up de estação	91
Figura 6.12	Precipitação mensal	91
Figura 6.13	Precipitação diária	92
Figura 6.14	Resultado de busca por nome	92
Figura 6.15	Aba detalhes de estação	92
Figura 7.1	Diferenças de testes manuais e automatizados	94
Figura 7.2	Esquema de sequência de login/registo no aplicativo	94
Figura 7.3	Composição de excerto de código e resultado dos testes com Vitest	95
Figura 7.4	Principal campo de atuação atual	96
Figura 7.5	Faixa etária	97
Figura 7.6	Gênero	97
Figura 7.7	Valores por itens agrupados	99
Figura 7.8	Qualidades atratividade, pragmática e hedônica	100
Figura 7.9	Resultados comparados com referências de estudos prévios	101

## ÍNDICE DE TABELAS

Tabela 2.1	Satélites com mensuração de variáveis hidrológicas	17
Tabela 2.2	Repositórios de dados hidrológicos	25
Tabela 2.3	Bibliotecas Javascript para camada de interface de usuário	35
Tabela 3.1	Atividades do Design Science Research	49
Tabela 3.2	Resultados potenciais de um projeto de pesquisa em Design Science	50
Tabela 3.3	Esquema para problemas de design	53
Tabela 3.4	Esquema de design aplicado ao projeto de aplicativo	53
Tabela 3.5	Estrutura das atividades do processo de engenharia de software	54
Tabela 5.1	Rotas (endpoints strapi)	80
Tabela 6.1	Técnicas de experiência de usuário	82
Tabela 6.2	Cenário de um usuário	84
Tabela 6.3	Excerto código componente Map.tsx	85
Tabela 7.1	Escalas Metodologia UEQ	97
Tabela 7.2	Valores por itens individuais	98
Tabela 7.3	Resultados comparados com referências de estudos prévios	100

## LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

AJAX	AJAX
ANA	Agência Nacional de Águas e Saneamento
ANEL	Agência Nacional de Energia Elétrica
API	Application Programming Interface
CEMADEN	Centro Nacional de Monitoramento e Alertas de Desastres Naturais
CMS	Content Management System
CMSA	China Manned Space Agency
CNES	Centre National d'Études Spatiales
CPRM	Companhia de Pesquisa de Recursos Minerais
CPTEC	Centro de Previsão de Tempo e Estudos Climáticos
CRUD	Create , Read , Update e Delete
CSS	Cascading Style Sheets
CSV	Comma Separated Value
DOM	Document Object Model
DSR	Design science research
EO	Earth Observations
EUA	Estados Unidos da América
ESA	Agência Espacial Europeia
EUMETSAT	Organização Europeia para a Exploração de Satélites Meteorológicos
FAIR	Findable, Accessible, Interoperable, Reusable
FOSS4G	Free and Open Source Software for Geoinformatics
GDAL	Geospatial Data Abstraction Library
GML	Geography Markup Language
GMT	Generic Mapping Tools
GPM	Global Precipitation Measurement
HDF	Hierarchical Data Format versão
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Ambiente de Desenvolvimento Integrado
IMERG	Integrated Multi-satélite Retrievals
INPE	Instituto Nacional de Pesquisas Espaciais
INEA	Instituto Estadual do Ambiente (Rio de Janeiro)
ISS	Estação Espacial Internacional
JAXA	Japan Aerospace Exploration Agency
JWT	JWT JSON Web Token
MPA	Aplicativos de múltiplas páginas
MVT	Mapbox Vector Tiles

NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
OGC	Open Geospatial Consortium
OSM	OpenStreetMap
PBF	Protocolbuffer Binary Format
PNRH	Política Nacional de Recursos Hídricos
RIA	Rich Internet Application
RHNR	Rede Hidrometeorológica Nacional de Referência
RWD	Responsive Web Design
SIG	SIG Sistemas de Informação Geográfica
SIG-WEB	Sistemas de Informação Geográfica Web
SINGREH	Sistema Nacional de Gerenciamento de Recursos Hídricos
SNIRH	Sistema Nacional de Informações sobre Recursos Hídricos
SPA	Single Page Application
TIN	Redes integradas trianguladas
TRMM	The Tropical Rainfall Measuring Mission
UEQ	User Experience Questionary
URL	Uniform Resource Locator,
UI	User Interface
UX	User Experience
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service

# Introdução

## 1.1 Contexto e motivação

Nas últimas décadas, geotecnologias, como, por exemplo, sistema de informações geográficas (SIG) e sensoriamento remoto, têm sido crescentemente empregadas por especialistas para adquirir e gerir dados de natureza geoespacial, envolvendo estruturação, armazenamento, visualização e análise.

As aplicações contemplam diversas temáticas, abrangendo as dimensões cultural, econômico e política, envolvendo os ramos socioeconômico e ambiental, entre outros. Assim, dados de natureza geofísica sobre a Terra são essenciais para o monitoramento, modelagem e gestão da água e do meio ambiente.

Em especial, para uma adequada gestão hídrico-ambiental, cabe reconhecer que o ciclo hidrológico, sob uma perspectiva global, abrange fundamentalmente três sistemas: (i) atmosférico; (ii) terrestre; e (iii) oceânico. Os processos físicos intervenientes são complexos, incorporando diversas escalas de análise, o que torna a compreensão do ciclo hidrológico um desafio científico há séculos, remontando o seu estudo a civilizações antigas, como, por exemplo, egípcia, hebraica, grega e romano-egípcia.

O impacto do ciclo hidrológico nas diversas sociedades, por sua vez, torna a demanda pelo uso de geotecnologias cada vez maior, como fica evidente, sobretudo nos últimos anos, diante da preocupação crescente da humanidade com mudanças climáticas e evolução da cobertura e uso do solo. Em maior ou menor grau, há influências antrópicas. e naturalmente preocupações com a continuidade da vida no planeta.

Para tanto, a resposta para diversas questões climáticas e existenciais, segundo o contexto apresentado, passa, em parte, por sistemas que permitam desenvolver atividades de coleta e manipulação de dados espaço-temporais complexos.

Por outro lado, os sistemas de gestão de informações requerem um nível de *expertise* em técnicas de geoprocessamento e tratamento sofisticado de dados, implicando, entre outras condicionantes, exigência de hardware com atributos de elevada complexidade tecnológica, e software do tipo SIG pagos, com demanda de uma curva de aprendizado para a sua utilização. Nesse sentido, grande parte das análises de natureza geocientífica necessitam

especialistas em geoprocessamento e uso de geotecnologias de forma mais ampla, mediante emprego de computadores robustos e softwares licenciados e pagos, localmente instalados, e uma estratégia de visualização de dados apropriada, o que constitui um desafio para as diferentes instituições e organismos da sociedade, delimitando claramente os limites de acessibilidade à informação por parte da sociedade.

O desafio do modelo de análise proposto e desenvolvido nas últimas décadas pela humanidade pode possivelmente ser mais bem compreendido, de forma mais evidente, quando enquadrado em três eixos principais: (i) grau de especialização necessário; (ii) hardware requisitado; e (iii) custo elevado de ferramentas pagas para realizar as análises.

Estabelecidas essas premissas, cabe observar que as dificuldades supramencionadas criaram uma espécie de abismo ou lacuna quanto à capacidade dos diversos setores econômicos da sociedade em lidar com a análise de informações geoespaciais para benefício ágil, rápido, preciso e seguro em processos de tomada de decisão diversos. Em meio a essa carência de acesso a informações em associação com a sua grande utilidade, prosperam grandes empresas, que faturam bilhões de dólares em venda de ferramentas, consultorias e análises geoespaciais customizadas.

Acrescente-se que, em meio a esse mercado de bilhões de dólares, a iminência de cenários ainda mais catastróficos relativos às questões climáticas emergentes e existenciais conectadas ao nexos água, energia, alimentação e ecossistemas, contribuem para que novas ferramentas computacionais poderosas e análises mais potentes e precisas recebam uma atenção de grandes empresas, incluindo o recolhimento de informações cada vez mais íntimas e extensivas de usuários, demarcando um território onde predominam disputas agressivas de marketing econômico, político e sobretudo cultural. Representam novos tempos com questões marcadas por uma reflexão aprofundada pelo tecido social, sendo este composto de forma distinta nas diferentes nações.

Por outro lado, a história aponta importantes iniciativas para servir de contrapeso ao que se convencionou chamar de mercado sob a ótica estritamente econômica, tendo, como contraponto, a democratização da informação e do conhecimento. Em outras palavras, com a adoção de um viés cultural, busca-se por meio de uma comunidade cada vez mais atuante e orientada, produzir ferramentas gratuitas e livres para disponibilizar um maior número de informações científicas perante diversos setores da sociedade, ampliando a acessibilidade para melhor gerir diferentes decisões de natureza pública e privada em distintos níveis.

De forma a ilustrar essa diretriz, no que concerne à área de sistemas de informações geográficas, destaca-se a iniciativa FOSS4G, que é uma sigla que se refere a "*Free and Open Source Software for Geospatial Applications*" ou "*Software Livre e de Código Aberto para Aplicações Geoespaciais*". O termo descreve um conjunto de softwares e tecnologias de código aberto que são usados para capturar, armazenar, analisar, gerenciar e visualizar dados geoespaciais, como mapas, dados de localização e informações relacionadas ao espaço geográfico. O FOSS4G engloba uma variedade de ferramentas e aplicativos de software livre que são amplamente usados em sistemas de informações geográficas (SIGs), cartografia, sensoriamento remoto e áreas correlatas. Alguns exemplos de software de natureza FOSS4G incluem QGIS<sup>1</sup>, PostGIS<sup>2</sup>, MapServer<sup>3</sup>, Geoserver<sup>4</sup>, MapLibre.

Em adição a esses esforços no desenvolvimento de ferramentas gratuitas, dois grandes desafios ainda persistem para realizar diferentes estudos e análises que poderiam beneficiar a sociedade de forma mais direta, rápida e justa. O primeiro deles diz respeito ao nível elevado de conhecimento que é necessário para compreender os diferentes tipos de dados livres que estão disponíveis, enquanto o segundo corresponde ao conhecimento de operar diferentes ferramentas livres, como, por exemplo, aquelas supracitadas associadas com a área de SIG. Mais especificamente, esse segundo desafio requer, muitas vezes, a integração numa aplicação única, em consonância com o custo computacional demandado em diferentes análises por parte dos usuários dessas ferramentas, buscando disponibilizá-las de forma a atender diferentes dispositivos com base no uso de tecnologias web modernas.

Em especial, pode-se afirmar que a presente dissertação propõe-se a contribuir diante da temática multidisciplinar que ambos os desafios explicitam.

Na solução desse duplo desafio, algumas iniciativas já têm sido apresentadas. Sob esse prisma, ganha força o uso da rede mundial de computadores, segundo a qual é possível, de forma mais democrática através de aplicações livres, disponibilizar a informação gratuitamente já de forma trabalhada e processada para análise mais direta por parte de distintos setores da sociedade.

---

<sup>1</sup> QGIS é um software livre com código-fonte aberto, multiplataforma de sistema de informação geográfica que permite a visualização, edição e análise de dados georreferenciados. <https://qgis.org/>

<sup>2</sup> O PostGIS é uma extensão espacial gratuita e de código fonte livre. Sua construção é feita sobre o sistema de gerenciamento de banco de dados objeto relacionado PostgreSQL, que permite o uso de objetos GIS serem armazenados em banco de dados. <https://postgis.net/>

<sup>3</sup> MapServer é um software livre que serve como ambiente de desenvolvimento para construção de aplicativos espaciais na internet. <https://mapserver.org/>

<sup>4</sup> GeoServer é um servidor de código aberto escrito em Java que permite aos usuários compartilhar, processar e editar dados geoespaciais. <https://geoserver.org/>

Reduz-se, assim, o hiato entre o conhecimento necessário e o custo computacional demandado, onde, apenas com o uso de computadores instalados num servidor, torna-se possível trabalhar com os dados e realizar análises de problemas que demandam soluções por diferentes instâncias da sociedade. Muitas dessas iniciativas têm sido apoiadas em usos de SIG-WEB, já customizados para a análise e interpretação de variáveis que concentram atenção em específicas temáticas. Adicionalmente, as iniciativas de reduzir a complexidade no uso das informações, possibilitando inclusive a visualização das análises e resultados de forma mais direta e intuitiva, têm sido estimuladas.

A presente dissertação, apoiada na visão de trazer a informação de maneira mais democrática, clara e acessível para sociedade, propõe desenvolver um sistema aberto, responsivo e disponível publicamente, onde a ênfase reside no princípio de que dados abertos podem ser visualizados de maneira direta e com fácil interpretação.

Diante do enorme desafio esboçado e das lacunas que se identificou na área temática em que se insere a pesquisa, empreenderam-se esforços para que a presente dissertação pudesse oferecer uma contribuição técnico-científica socialmente relevante.

## **1.2 Problema de investigação**

Observa-se, nas últimas décadas do século XX, notadamente a partir da década de 70, o incentivo acentuado ao lançamento de missões espaciais e satélites para a compreensão dos recursos naturais planetários. O avanço computacional e tecnológico produz uma vasta e crescente quantidade de dados hídricos e ambientais, demandando potência computacional ampliada embutida nas novas máquinas e recursos humanos altamente bem formados e qualificados para operar em face dessa profusão e diversificação de informações.

Por outro lado, ainda que haja uma quantidade significativa de dados, há uma dificuldade de acessá-los, o que leva, muitas vezes, à tomada de decisões limitadas, até certo ponto desinformadas ou não suficientemente embasadas, pelos diversos atores.

Para enfrentar o desafio enunciado, diversas ações são lançadas pelos órgãos governamentais e por instituições de pesquisas de diversos países para acesso, armazenamento, tratamento, modelagem e disponibilização de observações terrestres coletadas localmente ou adquiridas por meio de missões espaciais com vistas a produzir mapeamentos e modelagens diversas, como é caso de produção de campos pluviométricos, previsão numérica do tempo, modelagem hidrológica, evolução da cobertura e uso do solo,

entre outras aplicações.. Nesse sentido, enfatiza-se que a disponibilização da informação hídrico-ambiental requer um número de técnicas e artefatos informáticos, que serão objeto de estudo mais detalhado desta pesquisa.

Complementarmente, destaca-se que um fator que contribui para a dificuldade de acesso à informação e ao compartilhamento dos dados é seu o grande volume em associação com sua interpretação, traduzindo-se em elevados custos para pequenas organizações ou pessoas, o que determina a necessidade de ambientes, sempre que possível, mais integrados e dinâmicos.

Em particular, os dados hidrológicos geoespaciais são, muitas vezes, difíceis de serem assimilados, caracterizados e entendidos se forem apresentados estritamente como tabelas ou gráficos, modo usualmente empregado. São diversos os tipos de formatos de dados empregados em recursos hídricos, que, por sua vez, são frequentemente usados na tomada de decisões para permitir a análise das respostas de bacias hidrográficas em termos de vazão ou de impactos ambientais em função de medições pluviométricas bem como diante de cenários pluviométricos específicos, previstos ou projetados, com aplicações meteorológicas e climatológicas diversas.

No entanto, a maioria das partes interessadas e dos tomadores de decisão não dispõe do conhecimento técnico necessário para configurar ou desenvolver um software adequadamente para retratar ou simular um determinado conjunto de cenários que mereçam ser investigados. O processo torna-se ainda mais complexo para modelos hidrológicos lastreados em dados geoespaciais diversos, nomeadamente chuva, evaporação, umidade do solo, relevo e cobertura e uso do solo, entre outros, devido aos desafios de coleta de dados e gerenciamento de grandes conjuntos de dados espaciais e temporais com abrangências e escalas diversas (Sood & Smakhtin, 2015).

No que tange à utilização de softwares de código aberto, os benefícios em usar projetos de código aberto na área de SIG e recursos para atender às necessidades de aplicação web de recursos hídricos, geralmente, requerem combinar o uso de mais de um projeto ou *mashups*<sup>5</sup> para atingir a funcionalidade desejada. Os projetos de código aberto tendem a ser mais focados em termos de funcionalidade do que em suas contrapartes proprietárias. Enquanto os fornecedores de software proprietário normalmente oferecem uma ampla variedade de funcionalidades SIG, como por exemplo, serviços de mapeamento da web, geoprocessamento e armazenamento espacial em um único pacote de software, os projetos

---

<sup>5</sup> Um *mashup* é um site personalizado ou uma aplicação web que usa conteúdo de mais de uma fonte para criar um serviço completo.

de código aberto tendem a se concentrar em uma única categoria de funcionalidade. Conseqüentemente, a criação de uma aplicação web de recursos hídricos usando código aberto, via de regra, requer que o desenvolvedor sintetize vários projetos e gerencie o uso por meio de programação (Steiniger & Weibel, 2010).

A utilização de projetos de código aberto, oferecem inúmeras vantagens, mas também apresentam desafios específicos, principalmente ao integrá-los a sistemas existentes ou ao desenvolver novas soluções. Entre alguns desafios comuns estão: compatibilidade e padronização, manutenção e atualizações, documentação e suporte, personalização e flexibilidade. Neste sentido alguns projetos de código aberto podem não oferecer a flexibilidade necessária para atender a requisitos específicos de um sistema ou aplicação, exigindo modificações significativas. Para superar esses desafios, é importante avaliar cuidadosamente a seleção de projetos de código aberto, investir tempo na compreensão de sua arquitetura e documentação, e estar preparado para realizar adaptações e atualizações para garantir uma integração suave e eficaz.

No contexto do problema de investigação e temática da dissertação de tecnologias web, esse fato traduz-se na integração e no desenvolvimento de ferramentas do lado do servidor (*backend*) e do lado do cliente (*frontend*). Integrar essas tecnologias pode ser complexo devido à necessidade de comunicação eficaz entre as duas camadas. É necessário garantir que os dados fornecidos pelo servidor sejam compatíveis com o formato esperado pelo na camada de interface de usuário. Definir APIs consistentes para a comunicação entre as camadas também são cruciais para uma integração bem-sucedida. O backend e o frontend são geralmente desenvolvidos com linguagens e tecnologias diferentes. Garantir a compatibilidade e a comunicação adequada entre essas tecnologias torna-se um desafio a ser superado.

Um problema relacionado a ferramentas de código aberto na área de SIG é, normalmente, resultante da pouca atenção dedicada à interface de utilizador (UI) e ainda decorrente da experiência de utilizador (UX). As empresas privadas investem imensos recursos no refinamento e em melhorias de suas interfaces, o que não ocorre normalmente com projetos de código aberto, devido a limitação de orçamentos, equipe e tempo de desenvolvimento. O desenvolvimento *frontend* enfrenta diversos desafios em constante evolução devido às demandas por experiências de usuário mais ricas e complexas. Algumas dificuldades mais gerais enfrentadas são as compatibilidades entre navegadores e sistemas utilizados. Garantir que a interface funcione de maneira consistente em diferentes navegadores e dispositivos pode ser desafiador, pois cada um pode interpretar o código de maneira ligeiramente

diferente. A responsividade e a adaptabilidade também constituem desafios, pois a criação de interfaces que sejam responsivas e se adaptem a diferentes tamanhos de tela e dispositivos é essencial, mas pode ser complexo de implementar de forma a garantir uma experiência consistente.

A maioria das ferramentas SIG e FOSS4G<sup>6</sup> em sua interface de utilização ainda necessita de adaptações, refinamentos ou de uma camada UI que ainda não está implementada, e que demanda, de parte do desenvolvedor, uma escolha de quais componentes utilizar. Mesmo algumas soluções SIG em que existem uma camada de interface de usuário, ainda utilizam páginas tradicionais que são aplicações de múltiplas páginas, onde um novo documento HTML é carregado toda vez que o usuário altera a página ou seu conteúdo. Essa é uma opção relativamente lenta em comparação com a versão mais moderna de um modelo de desenvolvimento de aplicativo de página única (SPA), onde apenas as partes que foram alteradas são buscadas no servidor e atualizadas. A utilização de tecnologias modernas web pode reduzir a velocidade de carregamento de aplicativos, reduzir a utilização de recursos do servidor e melhorar a experiência do usuário.

### 1.3 Objetivos do projeto

Atualmente, existem vários conjuntos de dados, quer no formato de dados vetoriais (compostos de vértices e caminhos), quer no formato dito matricial (*raster*<sup>7</sup>). Os dados são originários de aquisições feitas por sensoriamento remoto, na forma espacial, ou a partir de sensores terrestres locais nas áreas de hidrologia, de meteorologia e de climatologia, segunda a forma denominada pontual. No caso da chuva, diversos produtos de satélite de chuva e dados observacionais de estações pluviométricas situadas na superfície terrestre permitem que a comunidade científica estime e quantifique a chuva precipitante e o correspondente impacto em bacias hidrográficas. No caso dos produtos de satélite, os dados de chuva numa cobertura quase em tempo real e global fornecem bases valiosas de dados aos cientistas para a construção de diferentes índices pluviométricos.

No entanto, conforme mencionado, a informação está direcionada a cientistas de dados, que são especialistas com capacidade de manipular grande quantidade de informação, em

---

<sup>6</sup> FOSS4G foi cunhado pela primeira vez no início de 2004 como um acrônimo para Software Livre e de Código Aberto para Geoinformática por um grupo de pesquisa que trabalha no I18N do GRASS e MapServer. <https://wiki.osgeo.org/wiki/FOSS4G>

<sup>7</sup> Em computação gráfica e fotografia digital, um gráfico *raster* representa uma imagem bidimensional como uma matriz retangular ou grade de pixels quadrados, também chamados de células de grade.

formato de dados não amigáveis de matrizes em múltiplas camadas. Além do conhecimento computacional, a complexidade apontada exige ferramentas e bom desempenho computacional, fatores que podem excluir grande parte da comunidade do acesso à informação.

Sendo gerados meios alternativos de disponibilização da informação, maior será o conjunto de pessoas que poderia usufruir dessa importante fonte de dados. Assim, os gestores e usuários com interesses diversos, como, por exemplo, na área de energia, construção e do agronegócio, poderiam ser beneficiados. Há o reconhecimento de que a informação sobre a chuva é, hoje, essencial nos mais diversos setores econômicos de uma sociedade bem como no planejamento dos afazeres no dia a dia das pessoas.

Com base no exposto, o objetivo principal deste projeto é desenvolver uma aplicação web responsiva de mapa temáticos que, usando fontes de código aberto, permita acessar, processar e visualizar dados sem que o usuário final precise de habilidades técnicas avançadas na gestão dos dados.

Os objetivos específicos para este projeto são:

1. identificar, selecionar e coletar dados a partir de repositórios abertos de informação hidrológica que suportem as necessidades de visualização de informação;
2. identificar, selecionar e instalar software que atenda ao espaço e a capacidades computacionais comumente requeridas para modelagem de recursos hídricos;
3. conceber e implementar uma arquitetura e uma infraestrutura de aplicação de projetos código aberto (servidor de bancos de dados, servidor de mapa, bibliotecas de interface da camada do utilizador) na área de SIG em suas diferentes interfaces de programação (APIs) do lado do servidor (*backend*);
4. desenvolver uma interface fluida e responsiva com a integração das diferentes (APIs) para diferentes dispositivos do lado do cliente (*frontend*);
5. disponibilizar um conjunto de mapas temáticos e métricas estatísticas em forma de gráficos significativos de informações hidrológicas através da aplicação web.

## 1.4 Metodologia de Investigação

O estudo utilizará um modelo de investigação definida como *Design Science Research* (DSR), que envolve o ato de aprender através do ato de construir. É uma metodologia em que o processo é iterativo. A DSR pode abordar de forma metodológica e científica a questão de se mostrar que a pesquisa que envolve a conceção de artefatos pode gerar conhecimento teórico. Além disso, a estrutura vertical do problema tratado ainda evidencia possíveis etapas que devem ser realizadas para a solução de cada um dos subproblemas que forem levantados.

De acordo com as diretrizes de Hevner *et al.* (2004), um projeto de investigação em formato DSR busca uma solução para um problema do mundo real de interesse para a prática.

A etapa do desenvolvimento é a efetiva construção do artefato utilizando os procedimentos de engenharia de *software*. Será utilizado um conjunto de artefatos necessários para solucionar o problema proposto. Para tanto, depende-se da utilização de várias técnicas em conformidade com o objeto que se está investigando.

Na etapa da avaliação, o artefato deve ser analisado e testado de acordo com as condições estabelecidas para validação; essa etapa contribui no processo de melhoria do artefato construído. Após a avaliação, segue o momento da conclusão, quando ocorre a análise e a interpretação dos resultados, possibilitando a consolidação do artefato.

## 1.5 Resultados Esperados

O estudo propõe uma abordagem apoiada num protótipo onde os usuários com diversos dispositivos informáticos poderão acessar as informações hidrológicas de interesse que tem normalmente necessidade de tratamento para serem visualizadas.

Utilizando-se de técnicas e *frameworks* ligados à metodologia *Responsive Web Design* (RWD)<sup>8</sup> e *Engenharia de Software*<sup>9</sup>, *User Experience*<sup>10</sup>, espera-se poder desenvolver uma aplicação web que possa ser rápido de carregar, sólido e atraente. Usuários, quer seja com

---

<sup>8</sup> *Responsive Web Design* (Web design responsivo) é a abordagem que sugere que o design e o desenvolvimento devem responder ao comportamento e ambiente do usuário com base no tamanho da tela, plataforma e orientação.

<sup>9</sup> Engenharia de software é uma área da engenharia e da computação voltada à especificação, desenvolvimento, manutenção e criação de software, com a aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade.

<sup>10</sup> A experiência do usuário, muitas vezes abreviada como “UX”, refere-se a como uma pessoa se sente ao interagir com um produto, sistema ou serviço. Abrange as percepções, emoções, preferências e satisfação geral do usuário durante e após sua interação.

sistema Windows em Desktop ou Notebook, quer seja também em dispositivos móveis Android ou iOS, com diferentes tamanhos de telas em smartphone, poderão utilizar o sistema sem restrições de funcionalidade. Espera-se construir um programa que atenda às necessidades de processamentos das informações hidrológicas e disponibilizar gráficos significativos de informação hidrológica através da interface de um mapa temático.

## **1.6 Organização do Documento**

Esta dissertação está organizada da seguinte forma. No Capítulo 1, discorre-se sobre a motivação, contexto e delineamento da questão científica proposta para o desenvolvimento da pesquisa.

No Capítulo 2, são apresentadas as bases conceituais e teóricas e conceitos utilizados, buscando-se ilustrar, com o suporte e referência a trabalhos, ferramentas e tecnologias relacionadas, a implementação concreta de sistemas web e de repositórios existentes que sejam relevantes para a elaboração desta dissertação.

Relativamente ao Capítulo 3, é apresentada a metodologia utilizada na dissertação, sobretudo com enfoque na etapa da metodologia de desenvolvimento do protótipo de software.

No Capítulo 4, após apresentação da metodologia, descrevem-se as entrevistas para especificações das funcionalidades do aplicativo, bem como as escolhas das tecnologias utilizadas para o tratamento dos dados e para o desenvolvimento da interface de usuário.

No Capítulo 5, são descritos o tratamento dos dados e a camada que abastecerá o aplicativo com os dados hidrológicos com base no acesso aos repositórios institucionais escolhidos.

No Capítulo 6, apresenta-se o desenvolvimento do protótipo com técnicas de experiência de usuário (UX) para criar e refinar a interface de utilizador do protótipo. As telas reais do protótipo com ênfase no design responsivo são discutidas.

O Capítulo 7 discorre sobre testes realizados, bem como também dispõe sobre os resultados do questionário de satisfação (experiência de usuário) aplicado a usuários de diferentes perfis.

Por último, no Capítulo 8, de modo a complementar a análise conduzida, no Capítulo 8, são tecidas as conclusões sobre os diversos elementos organizados e analisados ao longo do desenvolvimento desta dissertação, juntamente com a proposição de algumas recomendações para o desenvolvimento de oportunos trabalhos futuros, que levariam a melhorias do sistema desenvolvido em novos contextos que são naturalmente dinâmicos e adaptativos.

Neste capítulo, são apresentados tanto o estado da arte em relação à gestão de coleta, acesso e processamento de dados hidrológicos *in situ* e de satélites como também os trabalhos relacionados à área de visualização e disponibilização dessas informações mediante emprego de sistemas e plataformas computacionais com vistas a atender o público em geral. Na segunda parte do capítulo, são descritas, de forma geral, as tecnologias utilizadas para a disponibilização desses dados.

Na Seção 2.1, está inclusa uma síntese da coleta de observação pluviométrica, abrangendo instrumentos de medida de superfície, com destaque para a coleta *in situ* de dados de precipitação no Brasil, o uso do satélite para adquirir dados de precipitação, o processamento de imagens de precipitação por satélite, bem como as missões de satélite para coleta de precipitação e de variáveis intrinsecamente relacionadas.

Na Seção 2.2, são listados alguns repositórios de dados hidrológicos, suas interfaces, protocolos, formato de dados, resolução espacial e temporal. Em seguida, na Seção 2.3, refere-se às gerações de tecnologias de mapas na *web*, padrões e produtos de sistemas de informação geográfica.

Na Seção 2.4, descrevem-se as tecnologias gratuitas e de código aberto para área geoespacial como alternativas para as soluções fechadas e de custo elevado para indivíduos e pequenas organizações.

No que diz respeito à Seção 2.5, descrevem-se as algumas tecnologias e ferramentas pré-selecionadas com os objetivos de desenvolvimento de aplicação web.

Finalmente na Seção 2.6 descrevem-se alguns conceitos de Experiência de Usuário, Design de Experiência de Usuário e padrões de interface de usuário.

## 2.1 Coleta, acesso e observação de dados de precipitação

O avanço da ciência da água é criticamente dependente da recuperação e síntese de dados sobre a água. Central para esta tarefa são os dados de observações que descrevem as condições da água, como vazão, precipitação, qualidade e níveis das águas subterrâneas, medidos ao longo do tempo em diversos tipos de medidores e pontos de amostragem.

### 2.1.1 Monitoramentos de precipitação em superfície

As grandezas que caracterizam a precipitação são a altura (em mm ou cm) da lâmina d'água distribuída sobre a área atingida pela precipitação, a duração que pode ser de todo o evento ou estabelecida em intervalos pré-definidos dentro do evento, medida em minutos ou horas decorridas; a intensidade, que é a altura dividida pela duração e expressa em mm/h, e a frequência de ocorrência, que é uma característica estatística calculada, supondo-se a aleatoriedade do fenômeno.

A coleta, organização e distribuição de dados de água normalmente envolve investimentos maciços em infraestrutura técnica e mão de obra e utilizam diversos equipamentos para a coleta dos dados de precipitação. Os Registos de precipitação precisos e confiáveis são cruciais não apenas para o estudo das tendências e variabilidades climáticas, mas também para o gerenciamento de recursos hídricos e previsão do tempo, clima e hidrologia (Larson & Peck, 1974; Yilmaz *et al.*, 2005; Jiang *et al.*, 2012; Liu *et al.*, 2017, como citados em Sun *et al.*, 2018).

No caso da chuva, estações pluviométricas locais são utilizadas para oferecer diagnóstico e prognóstico sobre o excesso ou falta de chuvas, com enfoque no estudo de cheias e inundações bem como de vazantes e secas. A rede pluviométrica, no Brasil é, em várias regiões, esparsa ou inexistente. Quando existente, os postos que a compõem podem apresentar lacunas nos registros de dados, pois dependem de uma coleta regular por parte de observadores. Assim, análises estatísticas podem ser limitadas por falta de um registro longo e consistente, sem falhas.

Diante dessa limitação, as agências governamentais e instituições de pesquisa no Brasil, assim como as Universidades, têm buscado, desde o início dos anos 2000, investir no domínio e no uso de informações obtidas por sensoriamento remoto para complementar a rede de postos pluviométricas, identificando-se participação incipiente em programas espaciais. Aqui incluem-se produtos de chuva gerados por satélite, que têm permitido à comunidade científica produzir diversos trabalhos na área hídrica e ambiental.

Ainda assim, a complexidade envolvida no acesso e no tratamento de informações remotas oriundas de satélite abre espaço para novos desenvolvimentos na área de gestão da informação, incluindo nomeadamente as etapas de disponibilização e de visualização. A inserção de mais pessoas no processo de acesso a esse tipo de dado permite que se formulem planos de recursos hídricos mais adequados que permitam regular o uso da água com participação ativa da população. Projetos diversos na área de recursos hídricos e ambientais podem tornar-se mais apropriados e efetivos com o envolvimento da sociedade.

Entre as diretrizes que estão na Lei 9.433 de 9 de janeiro de 1997, conhecida, no Brasil, por “Lei das Águas”, estão a agência técnica de bacia e o comitê de bacia, que estão a crescer em termos de quantidade em todo o país, disseminando essa nova cultura do cuidado com a água. Na medida em que mais agências e comitês sejam formados para cuidar dos rios, a demanda da sociedade será crescente por informações mais acessíveis e dotadas de significado para tomadas de decisão na escala da bacia, que envolve entes governamentais, usuários e sociedade civil.

Um dos meios mais usados para os registos de precipitação são as estações pluviométricas instaladas em terreno, tipicamente usadas para medir diretamente a precipitação na superfície da Terra. Vários conjuntos de dados climáticos de grande escala em diferentes escalas espaço-temporais foram desenvolvidos a partir de observações de estação de superfície (*in situ*). No entanto, estas medições têm várias desvantagens, como cobertura de área incompleta e deficiências na maioria das áreas oceânicas e escassamente povoadas (Xie & Arkin, 1996; Rana *et al.*, 2015; Kidd *et al.*, 2017, como citados em Sun *et al.* 2018).

Dentre os equipamentos tradicionais e mais utilizados de registro contínuo, estão os pluviógrafos analógicos, os automáticos de báscula, os sensores óticos ou os de impacto resolvem a questão do registro ou amostragem temporal adequada ou real da precipitação. As formas de coleta podem ser os registos quase contínuos, com uma discretização mínima, usualmente de 5 minutos, mas pode ser que, em áreas urbanas, seja importante ter até 1 ou 2 minutos. Neste caso, os equipamentos de registro a cada acionamento do mecanismo de báscula, que tem uma capacidade máxima conhecida, poderão atender. Os mais difundidos são os pluviômetros para acumulação da precipitação e leitura por um observador a cada 24 horas.

Os pluviômetros fornecem medições de precipitação relativamente precisas e confiáveis em pontos únicos, mas são indisponíveis em muitas áreas oceânicas e escassamente povoadas e podem ser afetados por erros de amostragem. As observações de

satélite fornecem informações de precipitação com cobertura espacial homogênea, mas contêm erros aleatórios não desprezíveis e vieses devido à natureza indireta da relação entre as observações e a precipitação, amostragem inadequada e deficiências nos algoritmos. Muitas tentativas foram feitas para mesclar diferentes fontes de informação para superar esses problemas, aproveitando as vantagens individuais dos diferentes métodos, para obter análises de precipitação ideais com campos regulares em grade (Sun et al., 2018).



Figura 2.1: Equipamentos para medição de precipitação.

Fonte: Sun et al (2018)

Mais recentemente, dispositivos tecnologicamente sofisticados, como radares e desidrômetros, têm sido usados para aprimorar nosso conhecimento sobre a composição da precipitação e os potenciais processos físicos subjacentes à sua formação. Ao contrário dos pluviômetros, os disdrômetros podem detectar gotas de chuva individuais e medir seu tamanho. O conhecimento da distribuição do tamanho das gotas é essencial para entender os processos de precipitação, estimar a precipitação e melhorar as parametrizações microfísicas em modelos numéricos de nuvens.

Sobre a mediação por radar, este também pode capturar a estrutura tridimensional da precipitação. Eles são capazes de medir com mais realidade a distribuição temporal e espacial das chuvas. A vantagem mais importante do uso de radar para medições de precipitação é a cobertura de uma grande área com alta resolução espacial e temporal a partir de um único ponto e em tempo real. Além disso, a imagem tridimensional da situação meteorológica pode ser estendida a uma área muito grande pela composição de dados de vários radares. Também assim, as medidas obtidas com o radar são correlacionáveis com a precipitação que poderá ocorrer e podem ser usadas em modelos de previsão imediata da

precipitação sobre determinada região (Habib et al., 2012; Sauvageot, 1994, como citados em Sun et al., 2018).

### 2.1.2 *Uso de satélite para obter dados precipitação*

A tecnologia de sensoriamento remoto<sup>11</sup> é usado rotineiramente para obter informações precisas e oportunas para uma variedade significativa de aplicações, incluindo: o estudo do clima diário e mudanças climáticas de longo prazo; monitoramento do uso/cobertura do solo urbano-suburbano; modelagem ecossistêmica de vegetação, água, neve/gelo; agricultura; reconhecimento militar, e muitos outros usos. (Jensen, 2014)

Os Estados Unidos da América, Europa, China, Japão, Canadá, Índia e outras agências espaciais nacionais operam muitos sistemas de satélites que fornecem uma gama diversificada de tipos de medição e/ou cobertura espacial e temporal para a comunidade científica. Incluindo a Estação Espacial Internacional (ISS) e sistemas operados conjuntamente entre os EUA e agências internacionais, a NASA sozinha tem 18 grandes missões em ciências da Terra atualmente em órbita, enquanto a Agência Espacial Europeia (ESA) tem 11 missões de *Earth Observations*<sup>12</sup>(EO) em operação e uma série de futuras satélites em estágios avançados de planejamento e prontidão para lançamento (McCabe et al., 2017).

As últimas duas décadas testemunharam avanços surpreendentes em nossa capacidade de observar fenômenos hidrológicos, impulsionada em parte pela maturação do sensoriamento remoto por satélite, poder de computação crescente e capacidade de armazenamento de dados (Lettenmaier et al., 2015).

O sensoriamento remoto amadureceu como uma fonte viável de observações, particularmente em partes do mundo onde in situ as redes são escassas. Muitas variáveis hidrológicas podem ser estimadas através de sensoriamento remoto por satélite. Variáveis como medições globais de chuva, umidade do solo, cobertura de neve, mudança de armazenamento de águas subterrâneas, elevação de águas superficiais, e outras variáveis do ciclo da água dificilmente poderiam ter sido imaginadas quando a corrida ao espaço começou na década de 1960 (McCabe et al., 2017).

---

<sup>11</sup> Termo utilizado no Brasil. Em Portugal utiliza-se o termo Detecção Remota. É a medição ou aquisição de informações de alguma propriedade de um objeto ou fenômeno, por um dispositivo de gravação que não está em contato físico ou próximo com o objeto ou fenômeno em estudo.

<sup>12</sup> Earth observations: Observação da Terra é a coleta de informações sobre os sistemas físicos, químicos e biológicos da Terra via tecnologias de sensoriamento remoto, especialmente com satélites complementadas por técnicas de levantamento em campo, abrangendo a coleta, análise e apresentação de dados.

Uma lista de missões de satélites com variáveis hidrológicas pode ser visualizada na Tabela 2.1.

Tabela 2.1: Satélites com mensuração de variáveis hidrológicas

Variável Hidrológica	Missões/ Instrumentos	Resolução Espacial Padrão (km)	Resolução temporal padrão (dias)	Ano de Lançamento	Mensuração Dedicada
Chuvas	GPM	5	0.125	2014	Sim
Neve	GPM	5	0.125	2014	Não
Evaporação	Terra/MODIS	0.5	1	1999	Não
	Aqua/MODIS			2002	
	Suomi/VIIRS			2013	
	Landsat 8			2013	
	Landsta 9			2023	
Cobertura de Neve	Terra/MODIS	0.5	1	1999	Sim
	Aqua/MODIS			2002	
	Suomi/VIIRS			2013	
Densidade de Neve, profundidade, equivalente em água	GCOM-W/AMSR2	30	1	2012	Não
Umidade do solo superficial	SMOS	36	3	2009	Sim
	SMAP	36	3	2015	Sim
	ACAT	25	1	2006	Não
	GCOM-W/AMSR2	50	1	2012	Não
	Sentinel-1 <sup>a</sup>	0.1–0.005	12	2014	Não
	Sentinel-1B			2016	Não
Umidade profunda do solo	Biomass	0.2	18 dias ano <sup>-1</sup>	2021	Não
Elevação superficial da água	Jason-3	10	10	2016	Não
	SARAL 10 35 2013 N	10	35	2013	Não
	SWOT	0.1	11	2021	Sim
	ICESat-2	1.5	90	2018	Não
Mudanças no armazenamento de água terrestre	GRACE	220	30	2002	Sim
	GRACE-FO	180	30	2017	Não
Vegetação/cobertura do solo/área irrigada	Terra/MODIS	0.5	1	1999	Sim
	Aqua/MODIS			2002	
	Suomi/VIIRS			2013	
	Landsat 8	0.03	16	2013	Sim
	Landsta 9			2023	Sim
	Sentinel-2 <sup>a</sup>	0.02	10	2015	Sim
	Sentinel-2B	0.02	10	2017	Sim
	Sentinel-3 <sup>a</sup>	0.3	2	2016	Sim
Proba-V	0.35	2	2013	Sim	
Estresse da vegetação	ISS/ECOSTRESS	0.07	4	2018	Sim
Fotossíntese	FLEX	0.3	0.5	2022	Sim
Vapor de água	Aqua/AIRS	13.5	1	2002	Não

Fonte: M. F. McCabe et al., 2017: The future of Earth observation in hydrology

Especificamente para variáveis hidrológicas, até o momento, apenas um punhado de missões EO focadas em hidrologia terrestre foram projetadas e lançadas por agências espaciais nacionais e internacionais<sup>13</sup>. Isso foi possibilitado por uma mudança para missões mais orientadas ao usuário nas últimas duas décadas, o que permitiu aos cientistas pressionarem por suas necessidades (Lettenmaier et al., 2015).

Embora haja uma riqueza de dados sem precedentes que as missões de satélite tenham trazido grandes avanços no estudo da hidrologia em larga escala, permanecem lacunas que precisam ser preenchidas para aumentar nossa compreensão da hidrosfera, bem como questões que precisam ser abordadas para garantir o progresso contínuo do nosso conhecimento do sistema (McCabe et al., 2017).

Para melhorar essas lacunas, a precisão e cobertura da coleta de dados de precipitação, diversas missões de coleta de dados de sensoriamento remoto já foram lançadas nas últimas décadas. A geração atual de produtos pluviométricos multissatélites é amplamente baseada em algoritmos e atividades de validação centradas no TRMM (*The Tropical Rainfall Measuring Mission*) que foi um conjunto de satélites lançados pela NASA dos Estados Unidos e pela agência japonesa JAXA em 1997, que se concentra em chuvas médias a fortes sobre os oceanos tropicais (Hou et al, 2014).

Sendo anterior aos satélites da Tabela 2.1, a missão TRMM foi um conjunto de satélites de pesquisa em operação de 1997 a 2015, projetados para melhorar a compreensão da distribuição e variabilidade da precipitação nos trópicos como parte do ciclo da água no sistema climático atual. Ao cobrir as regiões tropicais e subtropicais da Terra, o TRMM forneceu informações muito necessárias sobre a precipitação e sua liberação de calor associada que ajuda a alimentar a circulação atmosférica global que molda o clima e o clima. Em coordenação com outros satélites do Sistema de Observação da Terra da NASA, o TRMM forneceu informações importantes de precipitação usando vários instrumentos espaciais para aumentar nossa compreensão das interações entre vapor de água, nuvens e precipitação, que são fundamentais para regular o clima da Terra (NASA, 2022).

Em seguida a missão TRMM, a NASA lançou a missão GPM (*Global Precipitation Measurement*) que veio a substituir a missão TRMM. A GPM foi lançada para fornecer uma nova geração de observações de chuva e neve do espaço para melhorar nossa compreensão do ciclo de água e energia da Terra. Como uma missão científica com objetivos de sistemas integrados, disponibiliza dados em tempo quase real (ou seja, dentro de 3 h de observação)

---

<sup>13</sup> Tabela 2.1. Variáveis hidrológicas e as missões de sensoriamento remoto por satélite atuais e planejadas que podem ser usadas para estimá-las.

para uma série de aplicações sociais que incluem a identificação de locais de tempestades, previsão de enchentes, monitoramento de água doce, alerta de deslizamentos de terra, previsão de safras e rastreamento de doenças transmitidas pela água. (Hou et al, 2014)

O conceito GPM centra-se na implantação de um satélite “*Core Observatory*” que transporta um sistema avançado de radar/radiômetro para medir a precipitação do espaço e servir como um padrão de referência para unificar as medições de precipitação de uma constelação de satélites de pesquisa e operacionais.

Iniciado pela NASA e pela *Japan Aerospace Exploration Agency* (JAXA) como sucessor global do TRMM, compreende um consórcio de agências espaciais internacionais, incluindo o *Centre National d'Études Spatiales* (CNES), a *Indian Space Research Organization* (ISRO), a Administração Nacional Oceânica e Atmosférica (NOAA), a Organização Europeia para a Exploração de Satélites Meteorológicos (EUMETSAT), entre outros. O resultado é o produto *Integrated Multi-satellite Retrievals* da missão GPM da NASA, chamado IMERG, que combina todos esses dados de 12 satélites em um mapa único e contínuo. A Figura 2.2 ilustra o conceito.

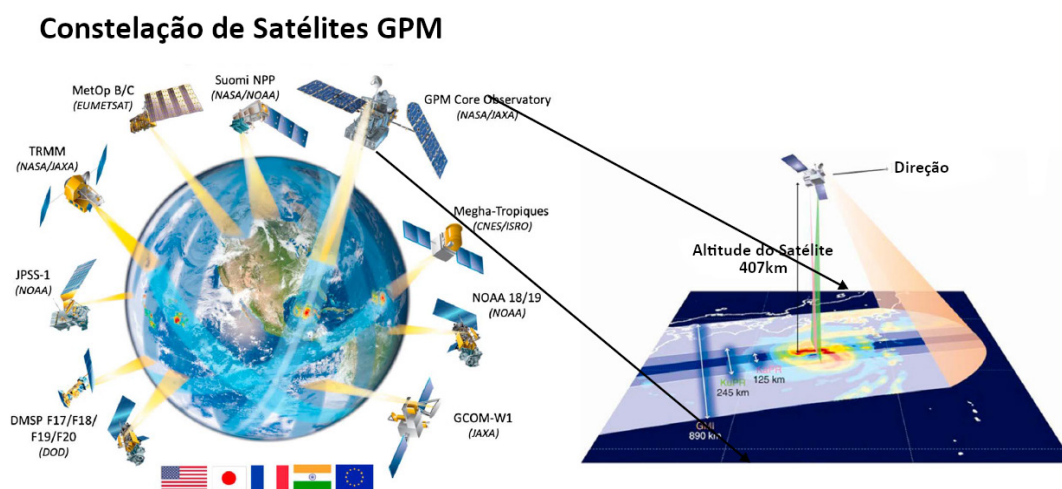


Figura 2.2: Conceituação da observação de precipitação da missão GPM

Fonte: adaptado de Hou et al. (2014).

### 2.1.3 Processamento de imagens de precipitação por Satélite

A medição hidrológica por satélite baseia-se em converter os eventos de precipitação em imagens, como matrizes, em que os níveis de cinza são convertidos para estimar a precipitação. Os dados de sensores remoto digital são geralmente armazenados como uma matriz de números. Cada valor digital está localizado em uma linha específica (i) e coluna (j) na matriz (Figura 2.3). Um pixel é definido como “uma imagem bidimensional elemento que é o menor elemento não divisível de uma imagem digital. Cada pixel na linha (i) e coluna

(j) na imagem tem um valor de brilho original (*Brightness Value*) associado com ele (alguns cientistas usam o termo número digital (*Digital Number*)). O conjunto de dados pode consistir em n bandas (k) de imagens multiespectrais<sup>14</sup> ou hiperespectrais<sup>15</sup>. Assim, é possível identificar o valor de brilho de uma pixel específico no conjunto de dados especificando sua linha (i), coluna (j) e coordenada de banda (k), isto é,  $BV_{i,j,k}$ . As n bandas são todas geometricamente relacionadas umas as outras. Portanto, uma entrada de interseção na banda 1 na linha 3, coluna 3 (ou seja,  $BV_{3,3,1}$ ) deve estar localizado na mesma coordenada de linha e coluna na quarta banda (ou seja,  $BV_{3,3,4}$ ). (Jensen, 2015).

### Dados em uma Imagem Digital de Sensoriamento Remoto (Raster)

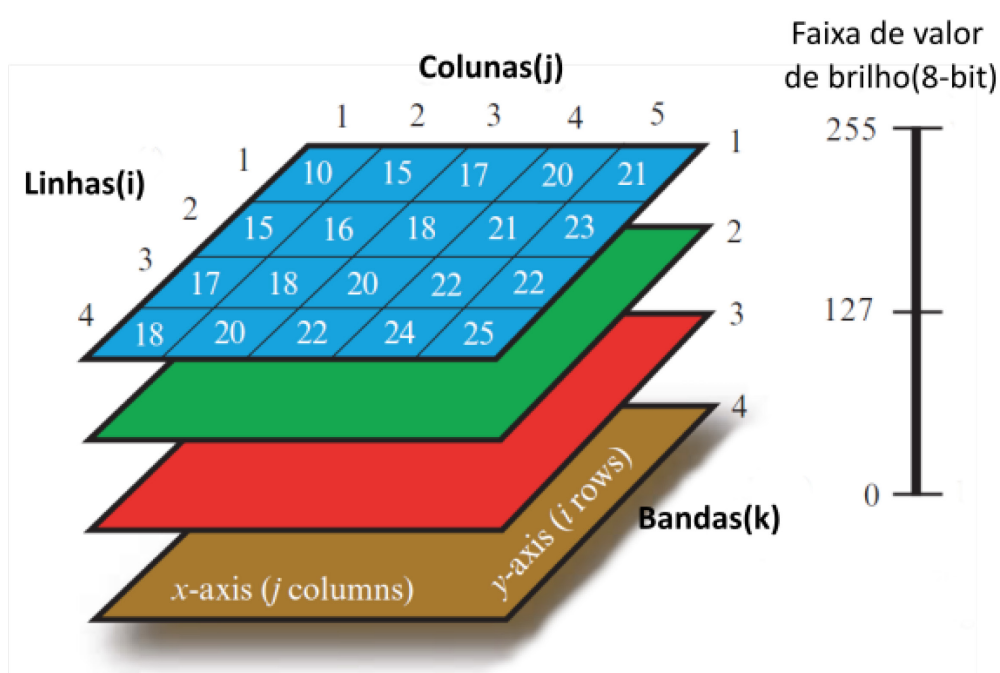


Figura 2.3: Imagem digital de sensoriamento remoto em camadas *Fonte: Jensen (2015)*.

As imagens de precipitação apresentam variabilidade em pequena escala que requer observações frequentes e espaçadas para uma representação adequada. Segundo a NASA (2022) está se tornando aparente que o futuro da pesquisa de precipitação provavelmente não será aquele em que as imagens e dados de satélite sejam usados isoladamente, em vez disso, a integração de medições de precipitação por satélite com observações terrestres como na figura 2.4 serão usados. Isso já é aparente nos produtos de precipitação analisados sobre

<sup>14</sup> Uma imagem multiespectral consiste em imagens de um mesmo objeto, tomadas com diferentes comprimentos de ondas eletromagnéticas. Pode ser luz visível, infravermelha, ultravioleta, raio-X ou qualquer outra faixa do espectro.

<sup>15</sup> A imagem hiperespectral coleta e processa informações de todo o espectro eletromagnético. O objetivo da imagem hiperespectral é obter o espectro para cada pixel na imagem de uma cena, com o propósito de encontrar objetos, identificar materiais ou detectar processos.

os EUA continentais e atividades semelhantes no Japão. Portanto, dentro desse contexto, as atividades de validação das imagens consideraram não apenas os produtos de satélite, mas os produtos de precipitação mesclados com base em dados em terra que também abrangem modelos de resolução de nuvens e modelos de resolução de superfície terrestre/nuvem acoplados usados em aplicações hidrológicas. Esses componentes de modelagem também são necessários para melhorar a formulação física do algoritmo.

A missão de satélites GPM mencionada anteriormente utiliza um algoritmo de validação chamado IMERG (*The Integrated Multi-satellite Retrievals for GPM*), que combina informação da constelação de satélites GPM para estimar precipitação sobre a maioria da superfície da terra. Este algoritmo é calibrado por algoritmos que mesclam e validam as medições de satélite com as medições na superfície, ele é particularmente valioso na maior parte da superfície da Terra que não possui instrumentos de medição de precipitação no solo (NASA, 2022). Particularmente em regiões oceânicas, remotas ou em desenvolvimento, embora seja possível criar tais estimativas apenas a partir de um tipo de sensor, os pesquisadores passaram cada vez mais a usar combinações de sensores e algoritmos de calibração, na tentativa de melhorar a precisão, cobertura e resolução (Huffman et al., 2007).

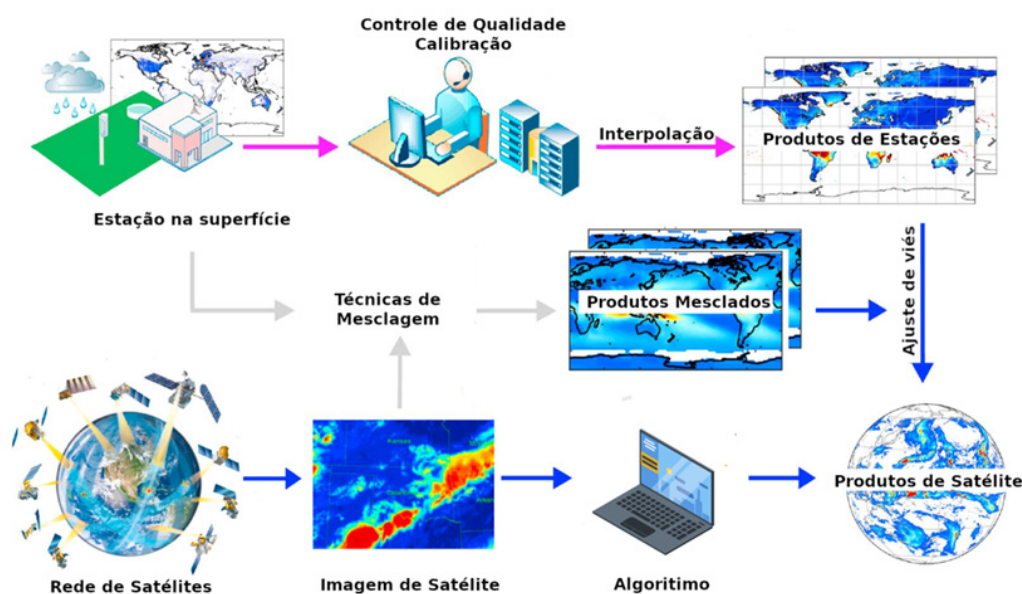


Figura 2.4: Fluxograma de calibração/validação de produtos de precipitação

Fonte: adaptado de Sun et al. (2018).

Um exemplo de produto de sensoriamento remoto é repositório de imagens chamado catálogo MERGE<sup>16</sup> que utiliza algoritmos de validação como descritos na figura 2.4, este catálogo/repositório é disponibilizado pelo Centro de Previsão de Tempo e Estudos Climáticos (CPTEC)<sup>17</sup>, uma divisão do Instituto Nacional de Pesquisas Espaciais (INPE)<sup>18</sup>, que mantém um banco de dados de aproximadamente mais de 1500 estações de superfície e utilizou as observações de satélite das Missões TRMM e GPM para produzir o produto mesclado MERGE (Rozante et al.,2010).

### *2.1.4 Coleta de dados de precipitação in situ no Brasil*

No Brasil, a Lei Federal no 9.433, de 08 de janeiro de 1997, instituiu a Política Nacional de Recursos Hídricos (PNRH) e criou o Sistema Nacional de Gerenciamento de Recursos Hídricos (SINGREH), além de tratar de outras regulamentações. Os dados gerados pelo SINGREH são incorporados ao Sistema Nacional de Informações sobre Recursos Hídricos (SNIRH). Dentre os instrumentos da Política Nacional de Recursos Hídricos, o art. 25 da referida lei estabelece que “o Sistema de Informações sobre Recursos Hídricos é um sistema de coleta, tratamento, armazenamento e recuperação de informações sobre recursos hídricos e fatores intervenientes em sua gestão”.

Dentre os esforços de atendimento aos objetivos do SNIRH, especificamente no que se refere à modelagem e ao tratamento dos dados geoespaciais da rede hidrográfica que compõe o banco de dados desse sistema, foi criada a Rede Hidrometeorológica Nacional (RHN), que é composta por Estações que monitoram volume de chuvas, o nível e a vazão dos rios, entre outros parâmetros. Inserida nela está a Rede Hidrometeorológica Nacional de Referência (RHNR), que engloba as estações cujos dados servem também a objetivos específicos, como gestão de risco de eventos críticos (cheias e secas) e gestão de recursos hídricos objetivando a proteção da vida e da propriedade e o avanço da pesquisa científica no Brasil. (Maranhão & Pereira, 2017). O fluxo de dados da coleta de dados hidrológicos no Brasil pode ser visualizado na Figura 2.5.

---

<sup>16</sup> <http://ftp.cptec.inpe.br/modelos/tempo/MERGE/>

<sup>17</sup> Centro de Previsão de Tempo e Estudos Climáticos - INPE - <https://www.cptec.inpe.br>

<sup>18</sup> INPE - Instituto Nacional de Pesquisas Espaciais - <http://www.inpe.br>

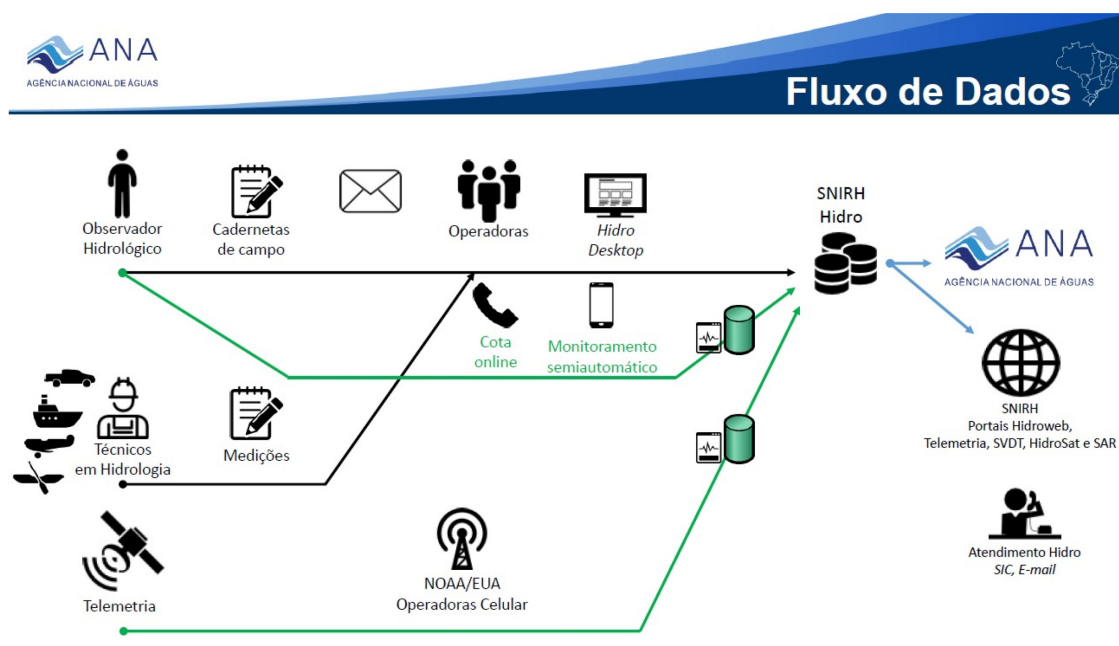


Figura 2.5: Fluxo de dados da rede hidrometeorológica nacional de referência

Fonte: Medeiros (2023)

A Agência Nacional de Águas e Saneamento (ANA), mantém e coordena as redes e monitoramento hidrológico do país, a RHN e a RNHR. Da totalidade das estações, 4.804 representam a rede básica nacional, em operação, especificamente de responsabilidade da Agência Nacional de Águas - ANA, constituída com o objetivo de avaliar a disponibilidade hídrica e conhecer o regime hidrológico das oito bacias hidrográficas brasileiras e sub-bacias constituintes (Maranhão & Pereira, 2017). Apenas em 2019 foram investidos R\$38 milhões. A principal parceria para operação da RHN é com a Companhia de Pesquisa de Recursos Minerais (CPRM) e envolve:

- 200 profissionais em campo que trabalham na instalação, manutenção e coleta de dados nas estações espalhadas em todo território;
- 3.500 observadores hidrológicos

O levantamento das informações necessárias à Hidrologia é realizado através de redes de estações hidrométricas, cujas séries de dados têm sua importância proporcional à sua extensão temporal. O cadastro de estações hidrométricas do País foi iniciado por volta de 1968, capitaneado pelo então Departamento Nacional de Águas e Energia Elétrica – DNAEE com publicações, mais tarde, dos primeiros Inventários de Estações Fluviométricas e de Estações Pluviométricas.

A ANA gerencia diretamente no âmbito da RHN, cerca de 4.528 estações, sendo: 1.816 estações fluviométricas; 111 estações fluviográficas; 2.712 pluviométricas; 280

estações pluviométricas; 485 sedimentométricas; e 1.611 estações de qualidade de água. A distribuição geográfica das estações é heterogênea, como se verifica na Figura 2.6. As estações de monitoramento supervisionadas pela ANA estão cadastradas num banco de dados denominado Sistema para Gerenciamento de Dados – HIDRO, que integra o Sistema Nacional de Informações sobre Recursos Hídricos – SNIRH<sup>19</sup>. A ANA é responsável pelo cadastro e inventário das estações e atribui, para cada uma, um código numérico de identificação, de oito dígitos, que leva em conta a bacia onde se situa a estação e sua posição nela. A operação e a manutenção das estações supervisionadas pela ANA são executadas por entidades públicas e privadas, com as quais são firmados termos de execução descentralizada de recursos, acordos de cooperação técnica ou contratos. (Maranhão & Pereira, 2017)

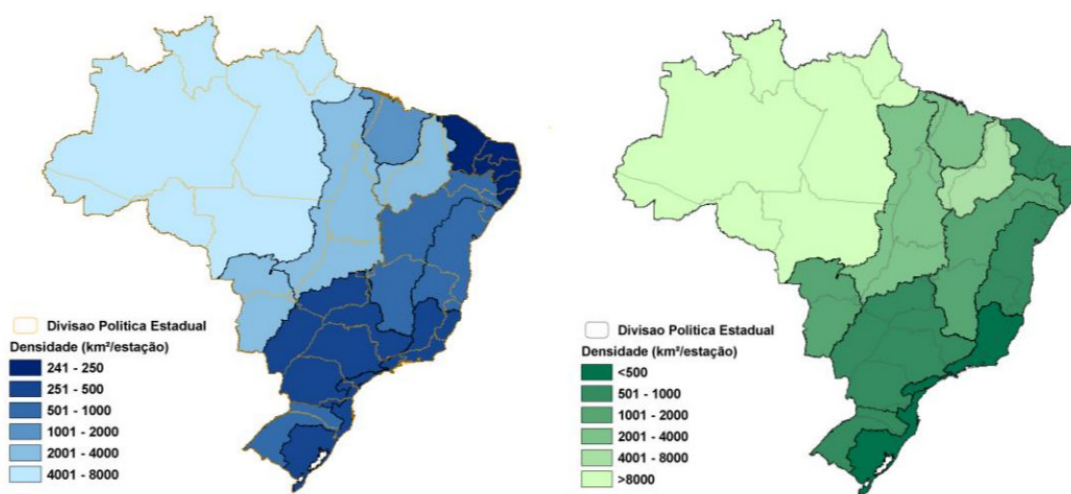


Figura 2.6: Densidade de estações pluviométricas (esq.) e fluviométricas (dir.) por região hidrográfica.

Fonte: Maranhão & Pereira (2017).

Uma das regiões que apresentam densidade maior de estações é o Nordeste semiárido, onde o conhecimento do comportamento da precipitação sempre foi uma preocupação. A rede é densa também nos estados do Sudeste e do Sul, onde o alto potencial hidrelétrico é explorado há mais tempo. Além das estações sob responsabilidade da ANA, também são integrantes da rede as estações mantidas pelos Estados no âmbito dos programas de implantação e operação das Salas de Situação Estaduais e do fomento ao monitoramento da qualidade da água.

<sup>19</sup> [www.snirh.gov.br](http://www.snirh.gov.br)

## 2.2 Repositórios de dados hidrológicos

Embora os pesquisadores e partes interessadas em geral possam acessar livremente grandes quantidades de dados ambientais fornecidos por organizações governamentais, agências distribuídas e grupos de pesquisa, geralmente na Internet, esses repositórios de dados online podem utilizar diferentes formatos de armazenamento e nomenclaturas para cada tipo de geociência.

Para atender às demandas de geoprocessamento em ambientes distribuídos como a Web, uma combinação de funções convencionais de análise e tecnologias avançadas de computação requerem infraestrutura técnica, modelos específicos de domínio. Arquiteturas, padrões e ferramentas para atender a esses requisitos foram sendo desenvolvidos paralelamente pelos diversos institutos de pesquisa, entidades governamentais, grupos de trabalhos online no mundo e estão sendo disponibilizados na web para serem aproveitadas para resolver os desafios de geoprocessamento. Essas organizações reconhecem que o acesso dinâmico a bancos de dados hidrológicos por meio de interfaces web padronizadas também facilitam o uso de modelos de simulação de dados intensivos e sistemas de apoio à decisão – obtendo assim o máximo benefício dos investimentos em coleta de dados. (Khatar, 2020)

Tabela 2.2: Repositórios de dados hidrológicos

Organização Repositório	Interface/ Protocolos*	Formato Dados**	Resolução Espacial	Resolução Temporal	Necessita Cadastro
NASA GPM - IMERGE <sup>20</sup>	HTTP-FTP	HDF5, GeoTIFF, NetCDF, OPeNDAP	Área, 0,10	30m*	Sim
INPE - MERGE <sup>21</sup>	HTTP-FTP	GRIB2	0,20	30m*	Não
ANA - Hidroweb <sup>22</sup>	WebMap, SOAP/REST	CSV, MDB, TXT	Pontual	Hora, Dia	Não
INEA – Alerta Cheias <sup>23</sup>	HTTP, Webmap	XLSX, CSV	Pontual	15m	Não
CEMADEN <sup>24</sup>	WebMap, HTTP	XLSX, CSV	Pontual	1h	Não
CPRM - SACE <sup>25</sup>	CMS, Webmap, HTTP	Tabela HTML	Pontual	15m	Não

<sup>20</sup> NASA GPM - <https://gpm.nasa.gov/data/directory>

<sup>21</sup> INPE - <http://ftp.cptec.inpe.br/modelos/tempo/MERGE/>

<sup>22</sup> ANA - <https://www.snirh.gov.br/hidroweb/serieshistoricas>

<sup>23</sup> INEA - <http://alertadecheias.inea.rj.gov.br/mapa.php>, O sistema de Alerta de Cheias foi criado pelo Instituto Estadual do Ambiente (INEA) para informar as autoridades e população sobre a possibilidade de chuvas intensas e inundações graduais (cheias) que possam causar perdas humanas e materiais.

<sup>24</sup> CEMADEN - <http://www2.cemaden.gov.br/mapainterativo>, Mapa Interativo da Rede Observacional para Monitoramento de Risco de Desastres Naturais do CEMADEN

<sup>25</sup> CPRM - <http://www.cprm.gov.br/sace> - O SACE (Sistema de Alerta de Eventos Críticos) é a plataforma desenvolvida pelo Serviço Geológico do Brasil (CPRM) para disponibilizar todas as informações geradas no contexto dos Sistemas de Alerta Hidrológico.

Dentre as diversas organizações e instituições que colaboram com dados abertos e repositórios disponibilizados para o público, selecionamos neste trabalho as seguintes: NASA (GPM Mission), ANA (Agência Nacional de Águas e Saneamento) do Brasil, INPE (Instituto Nacional de Pesquisas Espaciais) do Brasil.

### **\*Interface/Protocolos:**

- Http: Imagens Transferidas por protocolo HTTP
- Ftp: Imagens Transferidas por protocolo FTP
- WebMap: Interface de usuário com mapa dinâmico, onde o usuário pode selecionar o local de interesse para descarregar os dados.
- CMS (*Content Management System*): Portal web elaborado com opções avançadas de gerenciamento de usuários e conteúdo, com possibilidade de cadastro de usuários e permissões de acesso a conteúdo hierarquizados.

### **\*\*Formato de Dados**

- HDF5<sup>26</sup>: O *Hierarchical Data Format versão 5* (HDF5) é um formato de arquivo de código aberto que suporta dados grandes, complexos e heterogêneos. O HDF5 usa um "diretório de arquivos" como estrutura que permite organizar os dados dentro do arquivo de muitas maneiras estruturadas diferentes. O *Earth Observing System* (EOS) da NASA escolheu HDF como formato de dados e HDF, na forma de HDF5
- GeoTIFF<sup>27</sup>: O formato GeoTIFF foi desenvolvido inicialmente no início dos anos 90. O objetivo era melhorar um formato de arquivo independente de plataforma (TIFF) adicionando metadados necessários para descrever e usar dados de imagens geográficas
- NetCDF<sup>28</sup>: O NetCDF é um conjunto de bibliotecas de software e formatos de dados autodescritivos e independentes de máquina que suportam a criação, acesso e compartilhamento de dados científicos orientados a matrizes. As convenções para metadados de clima e previsão (CF) são projetadas para promover o processamento e compartilhamento de arquivos NetCDF. As convenções definem metadados que fornecem uma descrição definitiva do que os dados representam e as propriedades espaciais e temporais dos dados.

---

<sup>26</sup> HDF5: <https://www.hdfgroup.org/portfolio-item/earth-sciences/>

<sup>27</sup> GEOTIFF: <https://www.ogc.org/standards/geotiff>

<sup>28</sup> NETCDF: <https://www.ogc.org/standards/netcdf>

- OPeNDAP<sup>29</sup>: *Open-source Project for a Network Data Access Protocol* (OPeNDAP) é o desenvolvedor do software cliente/servidor, de mesmo nome, que permite aos cientistas compartilhar dados mais facilmente pela Internet. Com OPeNDAP, você pode acessar dados usando uma URL OPeNDAP de qualquer servidor de banco de dados que suporte OPeNDAP.
- GRIB2: GRIB é um formato de dados conciso comumente utilizado em meteorologia para armazenar dados históricos e meteorológicos de previsão. É padronizado pela Comissão de Sistemas Básicos da Organização Meteorológica Mundial<sup>30</sup>
- CSV: O formato de arquivo ("*Comma Separated Value*") que faz uma ordenação de bytes ou um formato de terminador de linha, separando valores com vírgulas é frequentemente usado para trocar dados entre programas diferentes.
- XLSX: Arquivo de planilha em formato XML aberto do Microsoft Office Excel

### 2.3 Padrões de Sistemas de Informação Geográfica

Existe uma grande heterogeneidade dos dados geoespaciais, e neste sentido *Open Geospatial Consortium*<sup>31</sup> (OGC) é uma organização internacional sem fins lucrativos, comprometida em melhorar o acesso a informações geoespaciais ou de localização, que representa mais de 500 empresas, agências governamentais, organizações de pesquisa e universidades unidas com o desejo de tornar as informações de localização *FAIR* (*Findable* -> Localizáveis; *Accessible* -> Acessíveis; *Interoperable* -> Interoperáveis; *Reusable* -> Reutilizáveis). (Open Geospatial Consortium Inc, 2020)

Os principais produtos de padrões da OGC são especificações de implementação e especificações abstratas. Esses padrões abertos da OGC permitem acesso e uso mais fáceis de informações geoespaciais e interoperabilidade aprimorada de tecnologias geoespaciais (em qualquer dispositivo, plataforma, sistema, rede ou empresa) para atender às necessidades da comunidade global. Interoperabilidade refere-se à capacidade de um produto de software de interagir com outros sistemas, devendo reduzir os custos de desenvolvimento, evitar a duplicação de esforços e prevenir o aprisionamento em apenas um fornecedor de tecnologia/software.

---

<sup>29</sup> OPeNDAP: <https://www.earthdata.nasa.gov/engage/open-data-services-and-software/api/opendap>

<sup>30</sup> <http://worldweather.wmo.int/pt/home.html>

<sup>31</sup> <https://www.ogc.org/>

Os padrões utilizados pela OGC são especializados de acordo com os tipos de dados que fornecem. Os mais importantes são o *Web Map Service* (WMS), o *Web Feature Service* (WFS) e o *Web Coverage Service* (WCS). O WMS produz mapas completos, renderizados no servidor e transmitidos como imagens ao cliente. Ele também pode responder a certas perguntas básicas sobre o conteúdo do mapa. O WFS fornece recursos individuais, codificados como vetores, selecionados do banco de dados espacial subjacente, para serem renderizados no lado do cliente. WCS fornece imagens e dados baseados em matriciais(*raster*) geográficos.

### 2.3.1 *Web Map Service (WMS)*

Web Map Service Interface Standard (WMS) fornece uma interface HTTP simples para solicitar imagens de mapas de um ou mais bancos de dados geoespaciais distribuídos. Uma solicitação WMS define a(s) camada(s) geográfica(s) e a área de interesse a ser processada. A resposta à solicitação é uma ou mais imagens de mapa georreferenciadas (retornadas como JPEG, PNG, etc.) que podem ser exibidas em um navegador web. A interface também suporta a capacidade de especificar se as imagens retornadas devem ser transparentes para que as camadas de vários servidores possam ser combinadas ou não.

### 2.3.2 *Feature Service (WFS)*

O WFS permite ao usuário fazer requisições de dados geoespaciais. Permite o retorno de geometrias vetoriais, bem como dados tabulares correspondentes. Define operações de interface da web para consultar e editar recursos geográficos vetoriais, como estradas ou contornos de lagos. Permite a análise, manipulação e edição de dados em nível GIS através da web. O WFS pode ter muitos recursos no servidor e na rede devido à quantidade de dados entregues. Enquanto um WMS retorna apenas uma imagem renderizada que geralmente é compactada e relativamente pequena, o WFS pode retornar grandes quantidades de dados no formato GML<sup>32</sup> (*Geography Markup Language*).

### 2.3.3 *Web Coverage Service (WCS)*

O WCS (*Web Coverage Service*) é um protocolo semelhante ao WFS, mas permite que um usuário interaja com dados espaciais e faça uma solicitação de dados matriciais(*raster*). Esse serviço é geralmente usado para dados de imagens de satélite especializados ou outros dados baseados em matriciais(*raster*) geográficos que podem conter

---

<sup>32</sup> GML é um formato de dados espaciais muito semelhante ao XML no layout.

informações por muitos anos e em diferentes contextos. O WCS tem uma filtragem mais poderosa do que o WMS e permite vários formatos de imagens e até dados multidimensionais, ou seja, imagens de vários anos que podem mostrar chuvas médias ou temperaturas ao longo de um período.

O *Web Coverage Service* (WCS) oferece suporte ao intercâmbio eletrônico de dados geoespaciais como "coberturas" – isto é, informações geoespaciais digitais representando fenômenos que variam no espaço. Exemplos de coberturas são imagens de satélite, modelos digitais de elevação e redes integradas trianguladas<sup>33</sup> (TINs). Um WCS fornece acesso a conjuntos potencialmente detalhados e ricos de informações geoespaciais, em formulários que são úteis para renderização do lado do cliente, coberturas de vários valores e entrada em modelos científicos. O WCS pode ser comparado ao *OGC Web Map Service* (WMS) e ao *Web Feature Service* (WFS). O WCS permite que os clientes escolham partes das informações de um servidor com base em restrições espaciais e outros critérios. Ao contrário do WMS, que filtra e retrata dados espaciais para retornar mapas estáticos (representados como imagens pelo servidor), o *Web Coverage Service* fornece dados disponíveis juntamente com suas descrições detalhadas; permite consultas complexas nesses dados; e retorna dados com sua semântica original (em vez de imagens) que podem ser interpretadas, extrapoladas etc.

### 2.4 Tecnologias de software gratuita e de código aberto para área geoespacial

Durante os últimos anos tem havido uma coalescência significativa de interesse em *Open Source Geospatial* (OSG), também conhecido e referido como *Free and Open Source for Geospatial* (FOSS4G). O uso do software proprietário (comercial) na área de sistemas de informação geográfica torna-se em geral inacessíveis para o usuário comum e para pequenas organizações devido aos custos altos. Alternativas viáveis, mas ricas em recursos, que pode ser atendida por software livre e de código aberto (Free and Open Source Software – FOSS).

---

<sup>33</sup> (TINs) são utilizadas pela comunidade GIS há muitos anos e são um meio digital para representar a morfologia da superfície. Os TINs são uma forma de dados geográficos digitais baseados em vetores e são construídos pela triangulação de um conjunto de vértices (pontos).

### 2.4.1 MapServer

O MapServer<sup>34</sup> emergiu da Administração Nacional de Aeronáutica e Espaço dos Estados Unidos da América (NASA) que patrocinou o desenvolvimento na Universidade de Minnesota para se tornar um dos pacotes SIG de código aberto mais populares disponíveis atualmente. O software fornece aos usuários as ferramentas básicas necessárias para construir aplicações web habilitadas espacialmente usando sua própria base de dados. O MapServer vem com um aplicativo de interface de gateway comum (CGI) pronto para uso que fornece funcionalidade para construir sites interativos, componentes Web 2.0 e instâncias de serviço do grupo Open Geospatial Consortium (OGC). Ele também possui MapScript<sup>35</sup>, uma poderosa interface de script para linguagens populares como PHP, C#, Java, Perl, Python e Ruby. O MapScript permite que os desenvolvedores adicionem funções geoespaciais a qualquer sistema (Hall, 2008).

### 2.4.2 GeoServer

Geoserver<sup>36</sup> é um servidor geoespacial de código aberto escrito em Java. Permite divulgar, gerenciar e analisar dados usando os padrões OGC mais utilizados (WMS, WFS, WCS e WPS), bem como extensões específicas para uma interação transparente com clientes como Google Earth ou outros softwares comerciais e Open Source. Ele também fornece suporte para protocolos padrão de fato como REST e formatos como GGeoJSON<sup>37</sup>. O GeoServer fornece as funcionalidades básicas para criar uma Infraestrutura de Dados Espaciais (SDI) interoperável de acordo com os padrões editados pelo Open Geospatial Consortium (OGC). O GeoServer foi criado para gerenciar e servir dados geoespaciais tanto vetoriais (chamado *features* pelo OGC) quanto raster (chamado *gridcoverage* pelo OGC), bem como para criar e divulgar mapas georreferenciados obtidos pela sobreposição de versões renderizadas dos dados mencionados acima de acordo com o estilo regras codificadas seguindo padrões específicos.

---

<sup>34</sup> <https://mapserver.org/>

<sup>35</sup> <https://www.mapserver.org/mapsript/>

<sup>36</sup> <https://geoserver.org/>

<sup>37</sup> O GeoJSON é um formato padrão aberto projetado para representar recursos geográficos simples, juntamente com seus atributos não espaciais. É baseado na JSON.

### 2.4.3 *OpenStreetMap, TileServer, OpenMapTiles*

O OpenStreetMap<sup>38</sup> (OSM) é um projeto colaborativo para criar um banco de dados geográfico editável gratuito. Os geodados subjacentes aos mapas são considerados a saída primária do projeto. A criação e o crescimento do OSM foram motivados por restrições no uso ou disponibilidade de dados de mapas em grande parte do mundo e o advento de dispositivos portáteis de navegação por satélite baratos. O OpenStreetMap é desenvolvido por uma comunidade voluntária de desenvolvedores que contribuem e mantêm atualizados os dados sobre estradas, trilhos, cafés, estações ferroviárias. O OSM é constituído por dados abertos: qualquer pessoa tem a liberdade de usar os dados para qualquer fim desde credite a autoria do OpenStreetMap e os seus contribuidores.

OpenMapTiles<sup>39</sup> e TileServer<sup>40</sup> são servidores de mapas em mosaico que geram imagens renderizadas de um banco de dados. É um conjunto de ferramentas de código aberto que você pode usar para gerar um servidor de mapas em mosaicos vetoriais (*Tiled Web Map*<sup>41</sup>) com dados do OpenStreetMap e de outros dados abertos ou seus próprios dados proprietários em um único mapa. Quando se olha para um “*Slippy Map*”<sup>42</sup> estamos interagindo com um conjunto de imagens individuais ou um conjunto de mosaicos individuais que são pré-gerados ou gerados sob demanda, e nos últimos anos estes mosaicos passaram de imagens matriciais (*rasters*) para imagens vetoriais, para que os dados vetoriais sejam transferidos para o cliente e renderizados sob demanda, com vantagens como cache para uma maior velocidade, que pode conter dados bruto para que o cliente possa alternar dinamicamente o idioma dependendo do navegador do visitante ou pode alterar as cores, o estilo dependendo da preferência.

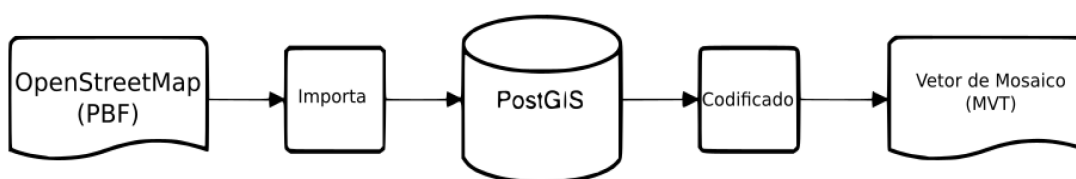


Figura 2.7: Como OSM é transformado em bloco de vetores.

<sup>38</sup> <https://www.openstreetmap.org/>

<sup>39</sup> <https://openmaptiles.org/>

<sup>40</sup> <http://tileserver.org/>

<sup>41</sup> Um mapa da web em mosaico, mapa deslizante ou mapa em mosaico é um mapa exibido em um navegador da web ao se juntar perfeitamente dezenas de imagens solicitadas individualmente ou arquivos de dados vetoriais.

<sup>42</sup> Slippy Map é, em geral, um termo que se refere a mapas da web modernos que permitem ampliar e deslocar (o mapa desliza quando você arrasta o mouse).

O projeto OpenMapTiles tem três recursos principais que definem a ligação entre as marcas(*tags*) do OSM e os dados brutos para gerar um mosaico vetorial e isso é chamado de esquema o que aparece em que nível de zoom quando você generaliza o conjunto de recursos e o conjunto de atributos que aparecem nos blocos de vetores finais que é o esquema de blocos de vetor.

Os ficheiros do OpenStreetMap (PBF<sup>43</sup>) são importados no banco de dados PostGis e codificados em mosaicos de vetores em ficheiros (MVT) para serem renderizados no navegador do cliente. Os blocos vetoriais são uma maneira de fornecer dados geográficos em pequenos pedaços para um navegador ou outro software cliente. Os blocos vetoriais (MVT) são semelhantes aos blocos *raster*, mas em vez de imagens matriciais, os dados retornados são uma representação vetorial dos recursos.

#### 2.4.4 Frameworks para camada de interface de usuários para Mapas

Existem *Frameworks*<sup>44</sup> para camada de interface de visualização para desenvolvedores que utilizam a linguagem Javascript que permitem adicionar interatividade aos mosaicos de imagens dos servidores de aplicação de mapas como GeoServer, Mapserver e ArcGis Server. As mais populares são OpenLayers<sup>45</sup>, Leaflet<sup>46</sup>, Mapbox<sup>47</sup> e MapLibre<sup>48</sup> sua contraparte de código aberto.

Leaflet é uma biblioteca em Javascript de código aberto para mapas interativos compatíveis com dispositivos móveis. É pequeno em tamanho e possui vários recursos que a maioria dos desenvolvedores precisa para mapas online. Suporta plataformas móveis e desktop.

Mapbox é um fornecedor de mapas online personalizados para sites que criou uma biblioteca de renderização em Javascript usada para criar mapas interativos usando WebGL<sup>49</sup>, que tanto permite desenvolver interface via frameworks de desenvolvimento web para

---

<sup>43</sup> O formato PBF ("Protocolbuffer Binary Format") destina-se principalmente como uma alternativa ao formato XML.

<sup>44</sup> Em português, o termo "framework" costuma ser utilizado sem tradução, sendo bastante difundido e compreendido na comunidade de desenvolvimento de software. No entanto, se for necessário buscar um equivalente em português, poderia ser utilizado o termo "estrutura de trabalho" ou "estrutura de desenvolvimento". Esses termos não são tão específicos quanto "framework", mas podem transmitir a ideia de uma estrutura ou conjunto de ferramentas pré-definidas para facilitar o desenvolvimento de software.

<sup>45</sup> <https://openlayers.org/>

<sup>46</sup> <https://leafletjs.com/>

<sup>47</sup> <https://www.mapbox.com/>

<sup>48</sup> <https://maplibre.org/>

<sup>49</sup> WebGL é uma API em JavaScript, disponível a partir do novo elemento canvas da HTML5, que oferece suporte para renderização de gráficos 2D e gráficos 3D. Pode ser implementado em uma aplicação web sem a necessidade de plug-ins no navegador.

*frontend*<sup>50</sup> como AngularJS<sup>51</sup>/Vue<sup>52</sup>/React<sup>53</sup> como também utilizar sua própria infraestrutura e seu aplicativo RIA<sup>54</sup> (*Rich Internet Application*) na nuvem chamada de Mapbox Studio. Após mudar sua licença para comercial criou-se uma divisão no desenvolvimento e a criação de sua contraparte de código livre chamada MapLibre.

Os componentes do MapLibre/MapBox giram em torno de estilos, *tilesets* e fonte de dados. A relação entre estes componentes pode ser visualizada na Figura 2.8.



Figura 2.8: Relação entre fontes de dados, *tilesets* e estilos.

Um *tileset* é uma maneira otimizada de salvar e transportar dados dividindo-os em tiles(blocos) blocos menores que são usados para criar imagens de mapas maiores. Um estilo define onde encontrar as fontes de dados (em forma bruta ou os *tilesets* otimizados) e como exibir esses dados. Isso permite que o usuário visualize cada bit de dados exatamente da maneira que deseja (ou não visualize: ocultando certas partes).

A linguagem de programação usada para construir o MapboxGLJS é JavaScript e WebGL (a versão web do OpenGL) é usada para renderizar o mapa dentro do navegador do cliente. A API de estilos e a API de mapas do Mapbox são usadas para determinar o que deve ser renderizado, como deve ser renderizado e recuperar os dados que precisam ser renderizados.

OpenLayers é outra biblioteca JavaScript de código aberto usada para exibir dados de mapas em navegadores da web. Ela também fornece uma API para criar aplicativos geográficos ricos baseados na web.

<sup>50</sup> O front-end é uma abstração, simplificando o componente subjacente pelo fornecimento de uma interface amigável.

<sup>51</sup> AngularJS é um *framework* web *frontend* de código aberto baseado em JavaScript para o desenvolvimento de aplicativos de página única. <https://angularjs.org/>

<sup>52</sup> Vue.js é um *framework* JavaScript de código-aberto, focado no desenvolvimento de interfaces de usuário e aplicativos de página única. <https://vuejs.org/>

<sup>53</sup> biblioteca JavaScript de código aberto com foco em criar interfaces de usuário (*frontend*) em páginas web. <https://react.dev/>

<sup>54</sup> RIA (Rich Internet Application) é um aplicativo da web que possui muitas das características do software de aplicativo de desktop.

Mapbox/Maplibre, OpenLayers e Leaflet suportam a integração de imagens através de *tilservers* (servidores de mapas e imagens/mosaico), como por exemplo para imagens de satélite. A renderização de imagens *raster* e dados de satélite nestas bibliotecas de camada de usuário normalmente necessitam algum plugin externo ou servidor de mapa de aplicação como GeoServer ou Mapserver, para que seja feito um pré-processamento para depois estas possam ser carregadas e visualizadas no mapa. Como exemplo temos na Figura 2.9, a biblioteca Openlayers a carregar um *raster* utilizando uma fonte externa.

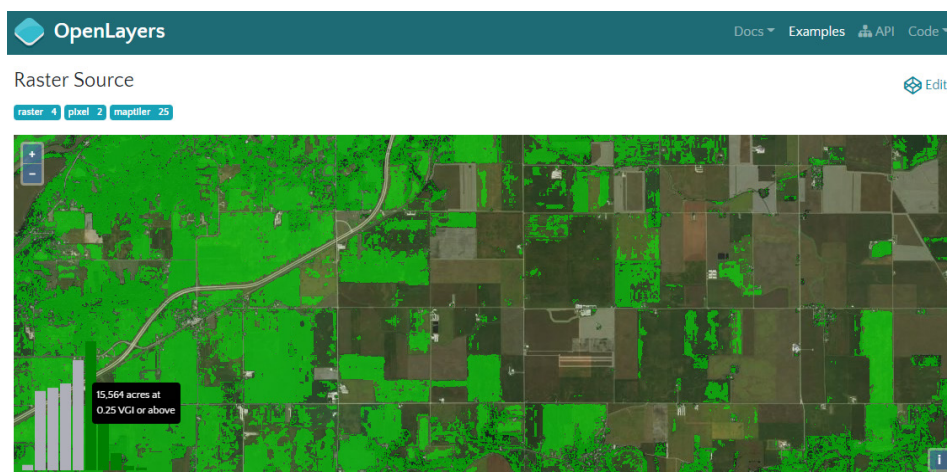


Figura 2.9: Openlayers, índice de verde da vegetação.

Fonte: OpenLayers (n.d.).

Todas as três bibliotecas suportam um mapa base padrão para uso imediato pelos usuários. OpenLayers e Leaflet usam o banco de dados do OpenStreetMap<sup>55</sup> como um de seus mapas base padrão. O Mapbox exige que os usuários se inscrevam em um *token*<sup>56</sup> de API para usar seu mapa base padrão e utilizar o servidor de mosaicos de mapas baseados nos dados do OSM. No entanto, os desenvolvedores podem usar livremente qualquer outro mapa base que não seja o padrão.

Estas bibliotecas além da interatividade com o clicar do mouse, também fornecem funcionalidades como a geocodificação, que é o processo de transformar uma descrição de um local — como um par de coordenadas, um endereço ou o nome de um lugar — em um local na superfície, inserindo uma descrição de local por vez ou fornecendo muitas delas de

---

<sup>55</sup> é um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo. <https://www.openstreetmap.org/>

<sup>56</sup> Um token de API é uma informação de autenticação usada para acessar uma interface de programação de aplicativo (API). Os tokens de API são usados como medida de segurança para garantir que apenas aplicativos ou usuários autorizados possam acessar recursos protegidos pela API.

uma só vez em uma tabela. As localizações resultantes são emitidas como feições geográficas com atributos, que podem ser usados para mapeamento ou análise espacial.

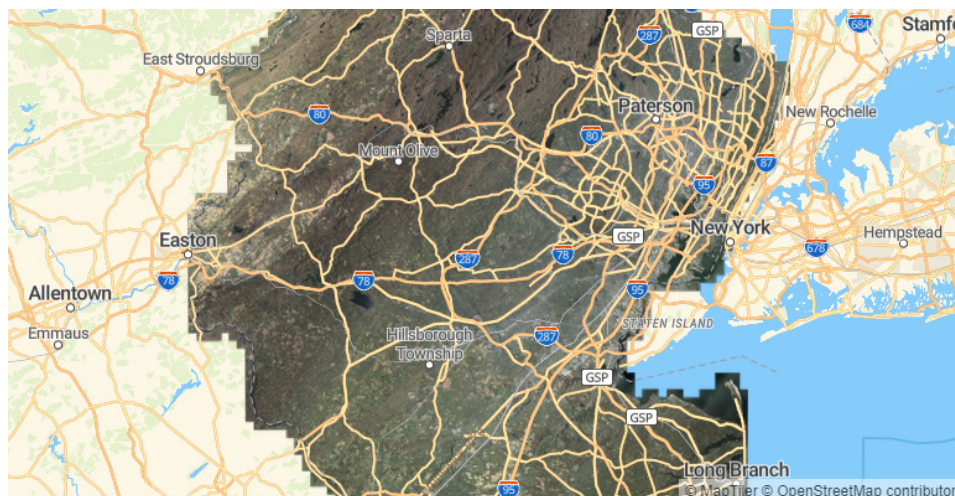


Figura 2.10: Camada raster Web Map Service (WMS).

O Mapbox/Maplibre suporta WMS (Web Mapping Service) e WFS, mas apenas mosaicos vetoriais no formato GeoJSON. OpenLayers suporta WMS e WFS. Leaflet suporta integração de Serviços OGC via plugin. Listamos algumas características destas bibliotecas na Tabela 2.3.

Tabela 2.3: Bibliotecas Javascript para camada de interface de usuário

Biblioteca	Padrões OGC	Formato Raster	Permite Uso de Mapa base próprio	GeoCodificação	Tipo de Licença
Leaflet	GeoJSON. *(CSV, WKT, TopoJSON, GPX)	plugin	Sim	plugin	BSD-2 <sup>57</sup>
OpenLayers	GeoJSON, ***(GeoRSS, KML, GML e WMS ou WFS).	extensão	Sim	extensão	BSD-2
Mapbox	GeoJSON, **	PNG, JPG, GeoTIFF (Nuvem)	Sim	API na nuvem	proprietária
Maplibre	GeoJSON	PNG, JPG GeoTIFF (Nuvem)	Sim	plugin	BSD

\*\* usando serviço na nuvem MapBox

\*\*\* usando servidores de aplicação como GeoServer ou Mapserver que servem através de sua API os recursos.

<sup>57</sup> Berkeley Software Distribution: Concede direitos de uso, incluindo o direito de relicenciar

## 2.5 Tecnologias de desenvolvimento na web

No desenvolvimento web, existem várias técnicas, tecnologias, linguagens de programação, banco de dados, frameworks e abordagens que os desenvolvedores utilizam para criar sites e aplicativos da web. Algumas das principais são: Tecnologias tradicionais como HTML (*Hypertext Markup Language*); CSS (*Cascading Style Sheets*), Javascript; Frameworks de interface do usuário (*React, Angular, Vue*); Frameworks “do lado do servidor” (*backend*) (*NodeJs, Ruby on Rails, Django, PHP, JAVA*).

Dentre as mais populares tecnologias atualmente empregadas, tem-se a JavaScript, que é uma das linguagens de programação para Internet mais utilizada no mundo, como mostra recentemente um estudo divulgado pelo sítio Stack Overflow<sup>58</sup> (2022) com mais de 71.000 desenvolvedores, que é mostrado na Figura 2.11:

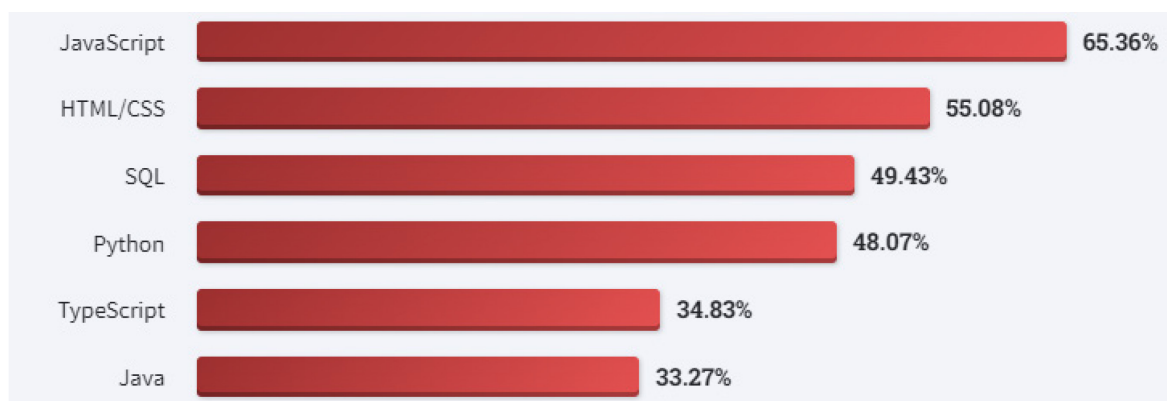


Figura 2.11: Linguagens mais utilizados pelos desenvolvedores.  
Fonte: Stackoverflow (2022).

Com esse aumento de uso e popularidade, era inevitável que uma grande quantidade de ferramentas e tecnologias fossem publicadas utilizando Javascript, sejam bibliotecas<sup>59</sup> ou *frameworks*.

Para desenvolvimento web em comparação com as bibliotecas, os *frameworks* de fato oferecem um conjunto completo de funções úteis e assume a responsabilidade por decisões que, de outra forma, o desenvolvedor teria que fazer antes de realmente escrever o

---

<sup>58</sup> *Stack Overflow* é uma plataforma online onde os desenvolvedores podem fazer e responder perguntas sobre programação. De acordo com sua página inicial, mais de 50 milhões de pessoas visitam seu sítio de todo mês. Link: <https://stackoverflow.com/>

<sup>59</sup> Para fins didáticos nesta dissertação, uma biblioteca geralmente consiste em código pré-escrito, classes, procedimentos, scripts, dados de configuração. Principalmente, pode ser integrada em um projeto existente com facilidade e usada para encurtar o tempo de desenvolvimento.

código do aplicativo. Isso inclui estratégias para roteamento no aplicativo, caminhos de URL, gerenciamento de estado, agrupamento de funcionalidades. (Wohlgethan, 2018)

Além disso, os *frameworks* fornecem melhorias no fluxo de trabalho que incluem as melhores práticas para aspectos básicos de desenvolvimento, como a estrutura geral de um aplicativo ou geração código *boilerplate*<sup>60</sup>. Os *frameworks* prometem maior produtividade e menor tempo ao mercado de desenvolvimento de aplicativos por meio de design e reutilização de código (do que com abordagens não baseadas nesta estrutura). Mas ainda assim, há situações em que a utilização de um framework pode resultar em uma sobrecarga em relação ao desenvolvimento. Por exemplo, conteúdo mais estático pode ser exibido com um site usual ou um Sistema de Gerenciamento de Conteúdo (CMS) como *WordPress*<sup>61</sup> (Wohlgethan, 2018).

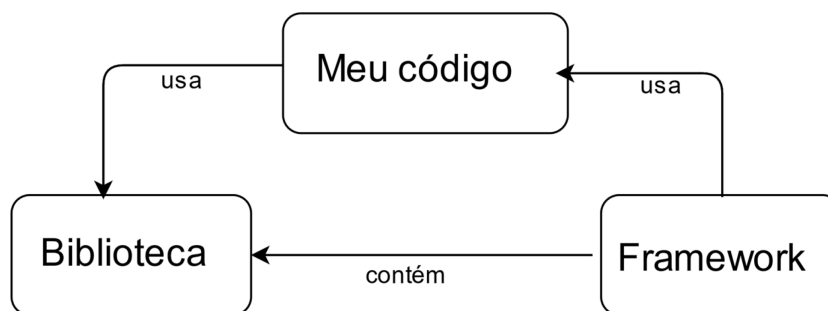


Figura 2.12: Diferença entre biblioteca e *framework*.

Fonte: adaptado de Wohlgethan (2018)

A seguir descrevem-se brevemente algumas das tecnologias que foram selecionadas para possível utilização no projeto do aplicativo.

### 2.5.1 Formas de desenvolvimento para Web

Existem muitas formas de desenvolver para a web. A escolha da abordagem depende dos requisitos do projeto, das metas de desempenho, do público-alvo e das preferências da equipe de desenvolvimento. Atualmente, existem duas formas de construir sites: como aplicativos de múltiplas páginas (MPAs) ou como aplicativos de página única (SPAs).

---

<sup>60</sup> Na programação de computadores, código *boilerplate* são seções de código que são repetidas em vários lugares com pouca ou nenhuma variação.

<sup>61</sup> <https://wordpress.org/>

Os aplicativos de página única são utilizados por diversas empresas para oferecer uma gama diversificada de serviços e produtos. Alguns exemplos mais notáveis incluem: Gmail, Facebook, Twitter, Google Maps, Trello, Airbnb, Slack.

Em seu início, as páginas da web tradicionais eram sítios de várias páginas, onde um novo documento HTML era carregado toda vez que o usuário alterava a página ou seu conteúdo. Essa é uma opção relativamente lenta em comparação com a versão mais moderna de um modelo de desenvolvimento de aplicativo de página única (SPA), onde apenas as partes que mudaram são buscadas no servidor e atualizadas.

Ao longo dos anos de utilização da web os desenvolvedores têm perseguido o objetivo de entregar aplicativos da web com a aparência de aplicativos de desktop. Várias soluções para uma experiência mais nativa, no final dos anos 90 e início dos anos 2000 foram utilizadas como Iframes<sup>62</sup>, miniaplicativos Java<sup>63</sup>, Adobe Flash<sup>64</sup> e Microsoft Silverlight<sup>65</sup> com algum grau de sucesso. Embora fossem tecnologias diferentes, todas elas tinham pelo menos um objetivo em comum: levar o poder de um aplicativo de desktop para o ambiente fino e multiplataforma de um navegador da web.

O aplicativo de página única (web), ou SPA, compartilha esse objetivo, mas sem um plugin de navegador. Essa ideia de experiência nativa evoluiu dos seus primeiros anos para atualmente ser aplicada usando apenas JavaScript, HTML e *Cascading Style Sheets* (CSS), sem a necessidade de instalar um plugin ou usar uma tecnologia proprietária. (Scott, 2015)

Essa metodologia nova de pensar sobre o design de páginas da web surgiu quando a tecnologia AJAX<sup>66</sup> começou a ganhar força. Tudo começou com um controle ActiveX no navegador Internet Explorer da Microsoft, começou a ser usado para enviar e receber dados de forma assíncrona. Quando a funcionalidade do controle foi oficialmente adotada pelos principais fornecedores de navegadores com a API XMLHttpRequest<sup>67</sup> (XHR), os

---

<sup>62</sup> Um iframe, abreviação de “quadro embutido”, é um elemento HTML usado para incorporar outro documento HTML no documento atual. Essencialmente, permite incluir conteúdo de uma página da web em outra.

<sup>63</sup> miniaplicativos Java ou também conhecidos como *applet* é um programa Java que pode ser incorporado em uma página da web. Ele roda dentro do navegador da web e funciona no lado do cliente. Um miniaplicativo é embutido em uma página HTML usando a tag APPLET ou OBJECT e hospedado em um servidor web.

<sup>64</sup> O Adobe Flash Player, ou apenas Flash, foi um reproduzidor de multimídia e aplicações amplamente distribuído.

<sup>65</sup> Microsoft Silverlight foi um software de tecnologia para navegadores e plugins, desenvolvido para competir com o Adobe Flash

<sup>66</sup> <https://developer.mozilla.org/pt-BR/docs/Web/Guide/AJAX>

<sup>67</sup> <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest> - Você pode recuperar dados de um URL sem precisar atualizar completamente a página. Isso permite que uma página da Web atualize apenas parte de uma página sem interromper o que o usuário está fazendo.

desenvolvedores que começaram a fundir esta API com JavaScript, HTML e CSS obtiveram resultados notáveis. A combinação dessas técnicas ficou conhecida como AJAX, ou “*Asynchronous JavaScript and XML*”. As solicitações de dados do AJAX, combinadas com o poder do JavaScript para atualizar dinamicamente o *Document Object Model* (DOM) e o uso de CSS para alterar o estilo da página em tempo real, trouxeram o AJAX para a vanguarda da tecnologia (Scott, 2015).

Aproveitando o sucesso e a popularização do AJAX, o conceito SPA leva o desenvolvimento web a um nível totalmente novo, expandindo as técnicas de manipulação de nível de página para todo o aplicativo. Além disso, os padrões e práticas comumente usados na criação de um SPA podem levar a eficiências gerais no design de aplicativos, manutenção de código e tempo de desenvolvimento se usados em conjunto com *frameworks*.

Os *frameworks* ditam o fluxo de trabalho de desenvolvimento de aplicativos, reduzem o tempo de desenvolvimento e possíveis erros. Cada *framework* é diferente com seus prós e contras, portanto, selecionar o correto para o projeto de software pode ser uma decisão estratégica para uma empresa e para o desenvolvedor júnior que busca adicionar uma nova habilidade ao seu arsenal. A partir de 2019, Angular, React e VueJS são algumas das estruturas JavaScript mais populares (Saks, 2019).

### 2.5.2 *Frameworks de Desenvolvimento em Javascript*

Na esteira do desenvolvimento de *Single Page Application* foram desenvolvidos diversos *frameworks* para facilitar o gerenciamento e criação das crescentes complexidades e funcionalidades de programação dos aplicativos webs. Nesta dissertação citamos três mais populares utilizados pelos desenvolvedores.

- **Angular:** foi originalmente criado pelos funcionários do Google Misko Hevery e Adam Abrons em 2008. Naquela época, era referenciado como AngularJS e desenvolvido em JavaScript simples. Isso ocorreu em uma época em que a maioria dos sites se baseava na abordagem de aplicativo de várias páginas: quando um usuário clicava em um link, o navegador precisava recuperar o documento HTML solicitado do servidor. Dependendo da conexão com a Internet e da capacidade de resposta do servidor, poderia levar um bom tempo até que o usuário pudesse visualizar a nova página. Gradualmente, os dispositivos aumentaram o desempenho geral para que a lógica do aplicativo pudesse ser executada no navegador. Isso levou à abordagem de

aplicativos de página única (SPAs). AngularJS foi um dos primeiros *frameworks* para o desenvolvimento de SPAs.

- **React:** O React foi desenvolvido em 2011 por Jordan Walke, engenheiro de software do Facebook. Este *framework* é atualmente desenvolvido pela equipe de desenvolvimento do Facebook e por uma comunidade de usuários individuais. A Meta Platforms<sup>68</sup> usa React em seus projetos, como Facebook e Instagram. O React foi projetado para aprimorar o desenvolvimento interativo da interface do usuário, facilitando a atualização da visualização quando os dados são alterados. Isso é feito dividindo a visão em componentes menores, que podem ser compostos para criar interfaces de usuários complexas. Os componentes são construídos em JavaScript em vez de modelos, permitindo um fluxo fácil de dados (Wohlgethan, 2018)
- **Vue:** Vue.js é *framework* em JavaScript leve, criada por Evan You. Vue é descrito como uma *framework* progressivo que pode ser usada para construir interfaces de usuário para a web. Embora não seja estritamente associado ao padrão *Model-View-Viewmodel*<sup>69</sup> (MVVM), os princípios de design do Vue foram parcialmente inspirados por ele. De acordo com seu site<sup>70</sup>, o *framework* pode ser usado tanto para pequenos projetos onde a biblioteca principal é usada entre outras tecnologias quanto para SPAs completos.

### 2.5.3 Responsive Web Design

Projetar um site alguns anos atrás costumava ser simples. Os usuários navegavam na web em computadores desktop. Embora os tamanhos dos monitores variassem um pouco, os sites projetados tinham um número limitado de resoluções. Smartphones e tablets forçaram os *programadores* a pensar fora dessa caixa. Atualmente, há uma grande variedade de resoluções nos diversos dispositivos que acedem aos portais na web (Glassman & Shen, 2014).

Para enfrentar o desafio de projetar portais para uma variedade de dispositivos, foi criada a abordagem de *Responsive Web Design*, que é uma abordagem que sugere que o design e o desenvolvimento devem responder ao comportamento e ambiente do usuário com base no tamanho da tela, plataforma e orientação.

---

<sup>68</sup> Meta Platforms, Inc empresa conhecida como Meta e anteriormente chamada Facebook, Inc.

<sup>69</sup> Model–view–viewmodel (MVVM) é um padrão de arquitetura de software que facilita a separação do desenvolvimento da interface gráfica do usuário (a visualização) – seja por meio de uma linguagem de marcação ou código GUI – do desenvolvimento da lógica de negócios ou back-lógica final (o modelo) para que a visualização não dependa de nenhuma plataforma de modelo específica.

<sup>70</sup> <https://vuejs.org/>

Menos de 10 anos atrás, era razoável construir um site com uma largura fixa. A expectativa era que todos os usuários finais tivessem uma experiência bastante consistente. Mas em 2007, o iPhone da Apple introduziu a primeira experiência de navegação por telefone verdadeiramente utilizável, e a forma como as pessoas interagem com a web mudou para sempre (Frain, 2020).

Segundo portal web statcounter.com, de novembro de 2012 até outubro de 2022, houve acentuada mudança na utilização de navegadores de Internet em relação aos dispositivos usados. Em outubro de 2022 a utilização em dispositivos móveis respondeu por 58.27%, enquanto nos computadores de mesa responde por 39.72%, e tablets apenas por 2%.

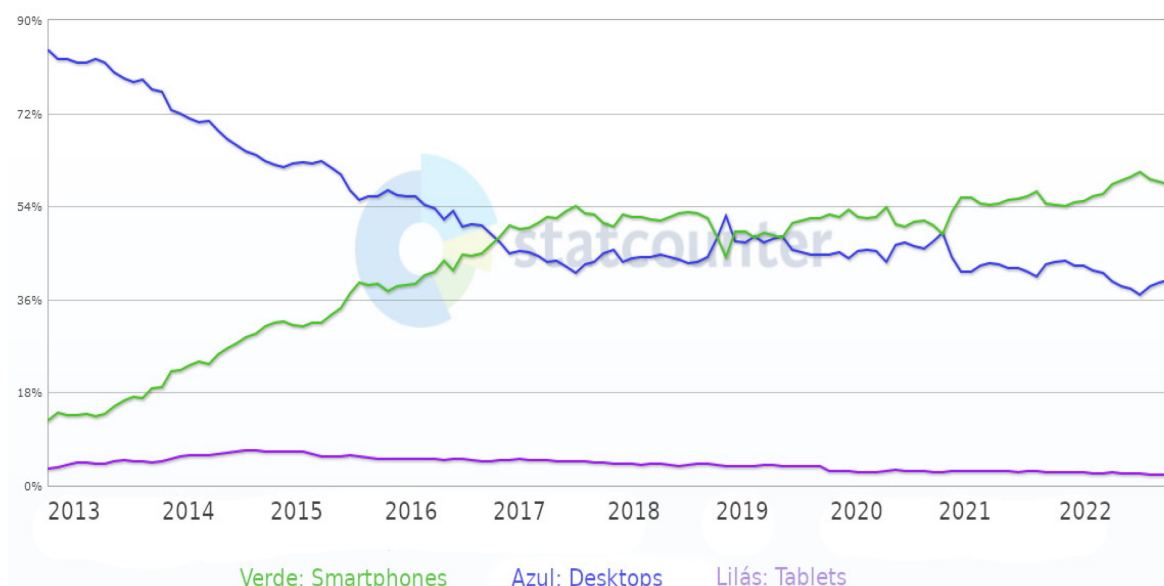


Figura 2.13: Navegadores por tipo de dispositivo

Fonte: Statcounter, 2022

Com a diversificação do acesso à Internet entre diferentes tipos de dispositivos e tamanhos de telas, houve necessidade de criação de mecanismos para responder e solucionar este problema. O termo "web design responsivo" foi cunhado por Ethan Marcotte em 2010. Em seu artigo seminal "A List Apart"<sup>71</sup> ele consolidou três técnicas existentes (layout de grade flexível, imagens flexíveis/ mídia e consultas de mídia) em uma abordagem unificada e nomeou-a *Responsive Web Design* (web design responsivo) (Frain, 2020).

O RWD basicamente permite que um site responda ou se adapte a um tamanho de *viewport*<sup>72</sup> (janela de visualização) diferente, menor ou maior, sem que você precise definir

<sup>71</sup> Marcotte, E. (2010). Responsive web design. A list apart.

<https://alistapart.com/article/responsive-web-design/>

<sup>72</sup> Uma **viewport** representa uma área poligonal (normalmente retangular) que está sendo exibida no momento. Em termos de navegador web, se refere a parte do documento que é exibida na janela

um domínio/subdomínio específico para pessoas que usam dispositivos móveis. A aparência do site pode ser mantida para ter experiências semelhantes em diferentes tamanhos de dispositivos. Isso é possível com o uso da *metatag*<sup>73</sup> *viewport* e “*media queries*”<sup>74</sup> CSS (consultas de mídia). (Firdaus, 2013)

Ter que desenvolver uma página da web com um público-alvo cada vez mais amplo é difícil, especialmente quando se tenta se adaptar a milhares de tipos diferentes de dispositivos (Lundmark, 2022). A utilização de frameworks específicos para implementação de RWD pode facilitar um pouco o trabalho dos desenvolvedores. Três frameworks populares são *Foundation*<sup>75</sup>, *Skeleton*<sup>76</sup> e *Bootstrap*<sup>77</sup>. Todos estes frameworks afirmam ser destinados para facilitarem o desenvolvimento web mantendo uma abordagem simples em relação à capacidade de responsividade e adaptação da visualização em navegadores com diferentes dispositivos.

Estes frameworks diferem em tamanho básico onde o *Skeleton* é um framework básico visando o controle total dos parâmetros, enquanto o *Bootstrap* é mais abrangente em opções e mais amigável em implementação, e o *Foundation* é um meio termo entre os dois primeiros.

O *Skeleton* é o framework mais leve dos três, com menos linhas de código, não destina-se a ser um pilar inicial para implementações de CSS e é baseado em um sistema de grade de 12 slots. O sistema de grade é a peça central do *Skeleton*, pois é uma estrutura mais leve com menos recursos ou funções extras (Karlsson, 2014).

O *Foundation* oferece uma estrutura com mais recursos avançados, como formatação de página da Web e e-mail, criações de menu responsivas e redimensionamento ou substituição automática de imagens. É uma estrutura mais abrangente, com mais complexidade para implementar. Este *framework* oferece suporte a um sistema de navegação responsivo que torna mais fácil para os desenvolvedores implementar menus responsivos. Os menus são redimensionados automaticamente dependendo da janela de visualização ou do tamanho do dispositivo e oferecem suporte à alteração da funcionalidade dependendo desses fatores (Karlsson, 2014).

---

<sup>73</sup> *metatag* é usada para definir as configurações da janela de visualização de uma página da web, especialmente para design responsivo.

<sup>74</sup> As consultas de mídia são um recurso do CSS (Cascading Style Sheets) que permite aplicar diferentes estilos a uma página da web com base em várias condições, como as características do dispositivo ou da janela de visualização. As consultas de mídia permitem que os desenvolvedores criem designs responsivos que se adaptam a diferentes tamanhos de tela, dispositivos e preferências do usuário.

<sup>75</sup> <https://get.foundation/>

<sup>76</sup> <http://getskeleton.com/>

<sup>77</sup> <https://getbootstrap.com/>

O terceiro *framework* selecionado é o *Bootstrap*, foi criada por desenvolvedores que trabalham sob a égide da equipe do Twitter<sup>78</sup> e é de tamanho semelhante ao Foundation. É um *framework* mais avançado, oferecendo diversas funções inteligentes e possibilidades para criar uma página web responsiva e eficiente. O sistema de grades oferece uma grade para fácil implementação de páginas responsivas. A grade é composta por doze colunas, onde o desenvolvedor tem total controle sobre como os elementos ocupam espaço e interagem entre si. Para imagens, existem classes auxiliares, que fazem com que a imagem responda de acordo com o tamanho do componente pai, fazendo com que a imagem seja reproduzida bem com redimensionamento de *tags div*<sup>79</sup>. As tabelas também oferecem suporte a classes auxiliares que tornam qualquer tabela maior que a largura da janela de visualização para poder rolar lateralmente (Karlsson, 2014).

### 2.6 Experiência de Usuário (UX) e Design UX

Podemos considerar a subseção anterior design responsivo como uma parte integrante do design de experiência do usuário, pois visa criar uma experiência consistente e eficaz em todos os dispositivos. O design de experiência do usuário, por sua vez, engloba uma gama mais ampla de considerações, incluindo a arquitetura da informação, a pesquisa do usuário, a usabilidade e a estética, além do aspecto específico do layout responsivo. Ambos trabalham juntos para garantir que os usuários tenham uma experiência positiva ao interagir com sites e aplicativos em qualquer dispositivo.

O design da experiência do usuário (UX) é o processo de aumentar a satisfação do usuário com um produto, criando uma interação utilizável, acessível e prazerosa entre o produto e seus usuários. (Bruce & Roger, 2020)

A UX pode ser implementada através de um ciclo de vida com uma série de atividades conectadas conforme a Figura 2.14.

---

<sup>78</sup> <https://twitter.com/>

<sup>79</sup> A tag/marca `<div>` é usada como um contêiner para elementos HTML - que são estilizados com CSS ou manipulados com JavaScript

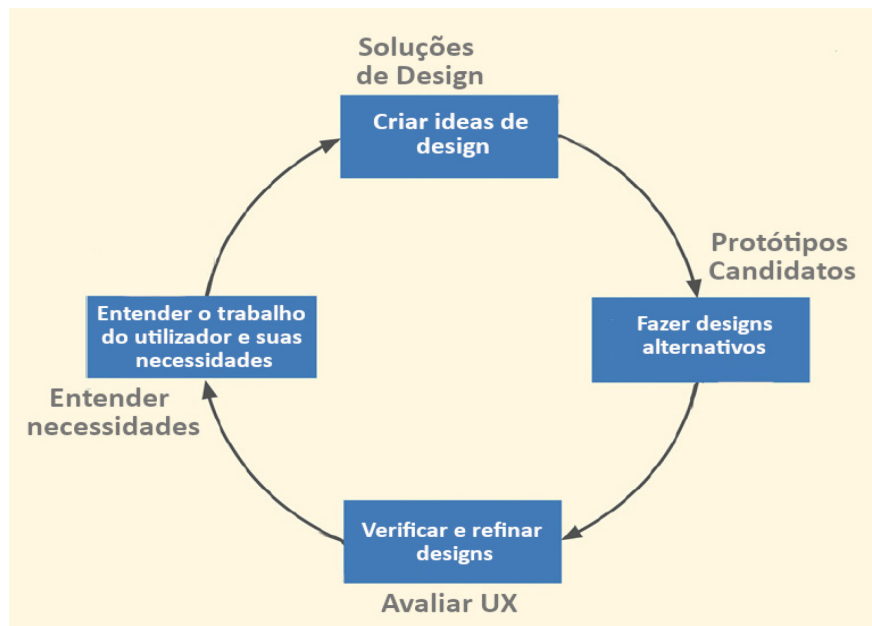


Figura 2.14: Ciclo de vida de atividades de design UX.

Fonte: adaptado de Hartson & Pyla (2018).

Para Hartson e Pyla (2018), o design de UX tem os seguintes passos:

- I. Entender as necessidades dos usuários.

**Métodos:** Elicitação de dados através de entrevistas e observações de usuários e coleta de dados sobre a prática de trabalho, uso e necessidades

- II. Projetar as soluções

**Métodos:** Ideação e esboço para criar ideias de design, criação de modelos mentais, modelos de sistema, storyboards, cenários ilustrados, *wireframes*<sup>80</sup> e *mockups*<sup>81</sup>

- III. Protótipos Candidatos

**Métodos:** Criação de protótipos funcionais que reúnem recursos visuais da etapa anterior e interatividade em uma versão do produto final.

- IV. Avaliação da Experiência do usuário

**Métodos:** Verificar e refinar o design UX para garantir que estamos a melhorar a experiência do usuário; avaliar o projeto com métodos empíricos ou analíticos para simular ou entender o uso real e produzir dados de avaliação; propor soluções de redesenho; reportar resultados.

<sup>80</sup> wireframe é um protótipo usado em design de interface para sugerir a estrutura de um site web e relacionamentos entre suas páginas.

<sup>81</sup> mockup é onde veremos os detalhes visuais, a aplicação de um layout, estampa, desenho, simulando o produto final.

Para orientar o processo de design, os projetistas precisam pensar sobre os elementos como personas, atividades, contexto e as tecnologias. As pessoas que irão utilizar o sistema são representadas por personas: perfis dos diferentes tipos, ou arquétipos, de pessoas para as quais o designer está projetando. As atividades e os contextos em que ocorrerão são vislumbrados por meio de cenários de uso (Benyon, 2019).

Um cenário de UX é uma história que descreve como um usuário pode interagir com um site, aplicativo ou produto em um contexto específico. É usado por designers para ajudá-los a entender e projetar para as necessidades e objetivos de seu público-alvo.

Em cenários e contextos estão contidas as *personas*. Uma persona em UX é um personagem fictício que representa um usuário de um site, aplicativo ou produto. Ele é criado por designers para ajudá-los a entender e projetar para as necessidades, objetivos e comportamento de seu público-alvo. Uma persona UX normalmente inclui detalhes sobre os dados demográficos, comportamentos, objetivos, pontos problemáticos do personagem e como eles podem interagir com o produto.

### 2.6.1 Padrões de Interface de Usuário

Os padrões de interface de usuário (UI) e o design de experiência do usuário (UX) estão intrinsecamente relacionados e são componentes fundamentais do processo de design de produtos e aplicativos digitais. Os padrões de UI podem influenciar o design responsivo, pois as diretrizes de design estabelecidas nos padrões de UI, como tamanhos de botões, espaçamento, tipografia e cores, precisam ser aplicadas de forma adaptável em diferentes tamanhos de tela.

Um padrão é uma solução recorrente para um problema em um contexto. Pode-se pensar em padrões como modelos: um padrão tem uma estrutura e pode ser facilmente usada para ajudar a resolver um problema mais rapidamente do que criar do zero.

Os padrões de interface do usuário são encontrados na esfera digital de sites, aplicativos, aplicativos móveis nativos e outros softwares ou dispositivos. Eles fornecem uma linguagem para pensar em um design interativo. Eles sugerem função, interação e intenção. Os padrões de interface do usuário documentam partes reutilizáveis de uma interface que compartilham um propósito (MacDonald, 2019).

Pode-se entender os padrões de UI explorando um *framework* de UI, Bootstrap descrito no item 2.7.1 com o exemplo do componente *Card*<sup>82</sup> mostrado na Figura 2.15.

---

<sup>82</sup> <https://getbootstrap.com/docs/4.0/components/card/>



Figura 2.15: Componente *Card* do *framework Bootstrap*.

Na Figura 2.15, o componente chamado *Card* tem o propósito de segmentar um conteúdo repetindo conteúdos para Imagens e Texto – sendo um padrão de visualização. Padrões de visualização fazem sentido em um *framework* de UI flexível que é construído com componentes.

Para MacDonald (2019), um padrão de UI é definido por três ingredientes:

- uma solução nomeada descrevendo o que o padrão faz;
- o problema que o usuário está enfrentando ou por que esse padrão é preciso;
- o contexto para quando usar o padrão.

Os padrões de IU comparam abordagens, descrevem as considerações e os sucessos dos designers anteriores. Conhecer os padrões e entender as decisões que os envolveram permite que se aproveite a crescente sabedoria de gerações e setores inteiros que criaram esses padrões, sem reinventar a roda. As soluções de IU pequenas e reutilizáveis encontradas nesses padrões podem ser compostas para criar experiências intuitivas e coesas que ressoam com as pessoas. (MacDonald, 2019)

Buscando padronizar e melhorar suas interfaces de usuário em seus aplicativos e páginas web, as grandes empresas de tecnologias desenvolveram e patrocinaram iniciativas, entre várias citamos as seguintes:

- **Material Design**<sup>83</sup> é uma linguagem de design desenvolvida pelo Google em 2014. Expandindo os "cards" que estrearam no Google Now, o Material Design usa layouts mais baseados em grade, animações e transições responsivas, preenchimento e profundidade efeitos como iluminação e sombras

---

<sup>83</sup> <https://m3.material.io/>

- **iOS Human Interface Guidelines<sup>84</sup> (HIG):** Desenvolvido pela Apple, é um conjunto de princípios de design para aplicativos iOS. Ele enfatiza a simplicidade, clareza e consistência na criação de aplicativos para dispositivos Apple.
- **Metro Design<sup>85</sup> (ou Fluent Design System):** Introduzido pela Microsoft, o Metro Design (agora conhecido como Fluent Design System) enfatiza layouts limpos, tipografia clara, uso de cores vibrantes e ícones simples e nítidos.

Uma iniciativa que implementa uma destes padrões é o *Material UI*, que é uma biblioteca de componentes React de código aberto que implementa o *Material Design* do Google. É abrangente e pode ser usado na produção imediatamente. Inclui uma coleção abrangente de componentes pré-construídos que estão prontos para uso na produção. O *Material UI* tem um design bonito e apresenta um conjunto de opções de personalização que facilitam a implementação de seu próprio sistema de design personalizado sobre seus componentes.

---

<sup>84</sup> <https://developer.apple.com/design/human-interface-guidelines>

<sup>85</sup> <https://metroui.org.ua/intro.html>

## Metodologia

Neste capítulo, expõe-se a metodologia proposta para a resolução do problema formulado na Seção 1.2.

Começando pela escolha do arcabouço estruturante para desenvolvimento da pesquisa (DSR), na Seção 3.1, apresenta-se a sua descrição geral, o projeto de artefatos, os ciclos de projeto, as partes interessadas e o contexto social. Em seguida, na Seção 3.2, é apresentada uma visão de seleção de técnicas e abordagens de engenharia de software, os requisitos de software, técnicas de elicitação dos requisitos, modelagem do sistema, o projeto de arquitetura, o projeto geral e a sua implementação bem como os componentes no nível de código e testes de software.

### 3.1 Design Research Science

Nesta dissertação a metodologia de pesquisa utilizada foi a *Design Science Research* (DSR) e para a produção do artefato em conjunção com a DSR, foram utilizadas metodologias de Engenharia de Software. Nesta seção descrevem-se as metodologias e procedimentos aplicados na dissertação e no projeto de software.

Segundo Wieringa (2014) para fazer um projeto de Design Science você precisa entender seus principais componentes a saber: seu objeto de estudo e suas duas principais atividades. O objeto de estudo é um artefato inserido em um contexto, e suas duas principais atividades são projetar e investigar este artefato no contexto particular.

Hevner et al. (2004) afirma que DSR pode ser entendida como um projeto e validação de planejamento de soluções para problemas práticos. A DSR pode e deve ser apresentada efetivamente para públicos voltados para gestão e tecnologia.

Para Hevner e Chatterje (2010), o DSR inclui seis etapas: identificação do problema e motivação; definição dos objetivos para uma solução, projeto e desenvolvimento; demonstração; avaliação; e comunicação. Para atingir os objetivos de pesquisa podemos identificar estes passos como atividades na Tabela 3.1.

Tabela 3.1: Atividades do *Design Science Research*

Atividades	Descrição
Identificação de problemas e motivação	Defina a pesquisa específica problema e justificar o valor de uma solução.
Defina os objetivos de uma solução	Inferir os objetivos de uma solução a partir da definição do problema e do conhecimento do que é possível e factível.
Design e desenvolvimento	Crie o artefato. Tais artefatos são constructos potenciais, modelos, métodos ou instanciações
Demonstração	Demonstrar o uso do artefato para resolver uma ou mais instâncias do problema.
Avaliação	Observe e meça quão bem o artefato suporta uma solução para o problema.
Comunicação	Comunicar o problema e sua importância, o artefato, sua utilidade e novidade, o rigor de seu design e sua eficácia para pesquisadores e outros públicos relevantes, como profissionais atuantes, quando apropriado.

Fonte: adaptado de Hevner & Chatterjee (2010)

Para Vaishnavi e Kuechler (2015) os resultados da DSR podem ser constructos, modelos, arquiteturas, e princípios de design, métodos, instanciações. DSR é o *design* e investigação de artefatos em um contexto. Os artefatos que estudamos são projetados para interagir com um contexto de um problema a fim de melhorar algo nesse contexto. O artefato

é algo criado para algum objetivo prático. Exemplos de artefatos projetados e estudados em sistemas de informação e engenharia de software são algoritmos, métodos, notações, técnicas e frameworks. (Hevner et al. 2004)

Tabela 3.2: Resultados potenciais de um projeto de pesquisa em Design Science

	Resultados	Descrição
1	Constructo	O vocabulário conceitual de um domínio
2	Modelos	Conjuntos de proposições ou declarações que expressam relacionamentos entre construções
3	Estruturas	Guias reais ou conceituais para servir de suporte ou guia
4	Arquiteturas	Estruturas de sistemas de alto nível
5	Princípios de Projetos	Princípios e conceitos fundamentais para orientar o design
6	Métodos	Conjuntos de etapas usadas para executar tarefas - conhecimento de como fazer
7	Instanciações	Implementações situadas em determinados ambientes que operacionalizam ou não construções, modelos, métodos e outros artefatos abstratos; neste último caso, tal conhecimento permanece tácito
8	Projeto teorias	Um conjunto prescritivo de declarações sobre como fazer algo para atingir um determinado objetivo. Uma teoria geralmente inclui outros artefatos abstratos, como construções, modelos, estruturas, arquiteturas, princípios de design e métodos.

Fonte: adaptado de Vaishnavi & Kuechler (2015).

Segundo Wieringa (2014), o conceito de artefato deve ser entendido de forma ampla, incluindo algoritmos e métodos, considerando até mesmo estruturas conceituais como ferramentas para a mente, que podem ser utilizáveis e úteis para propósitos particulares.

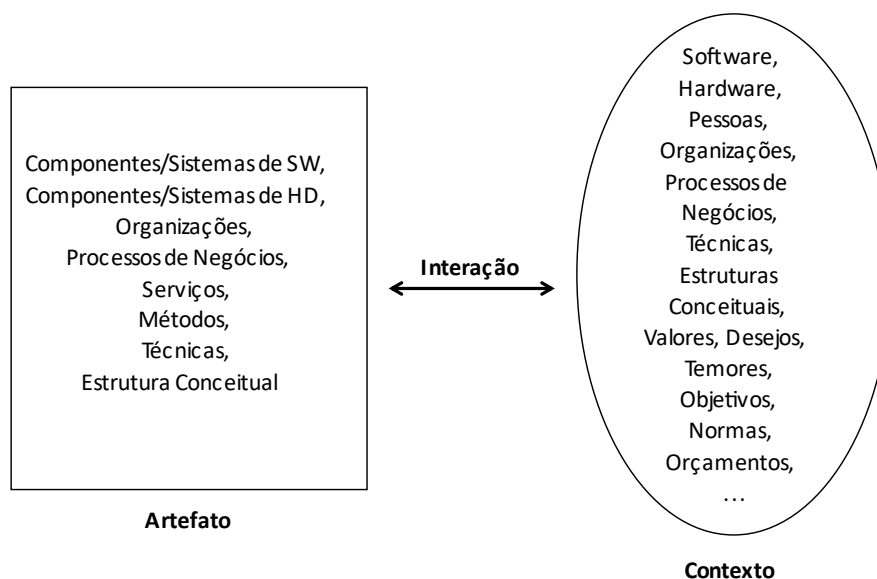


Figura 3.1: Artefatos e Contexto.

Fonte: adaptado de Wieringa (2014).

### 3.1.1 Ciclos de engenharia

Wieringa (2014) afirma que m projeto de ciência do design itera sobre as atividades de projetar e investigar. A tarefa de projeto em si é decomposta em três tarefas (Figura 3.2): investigação do problema, projeto do tratamento e validação do tratamento. Este conjunto de três tarefas, são ciclos de projeto, porque os pesquisadores repetem essas tarefas muitas vezes em um projeto de pesquisa em DSR.

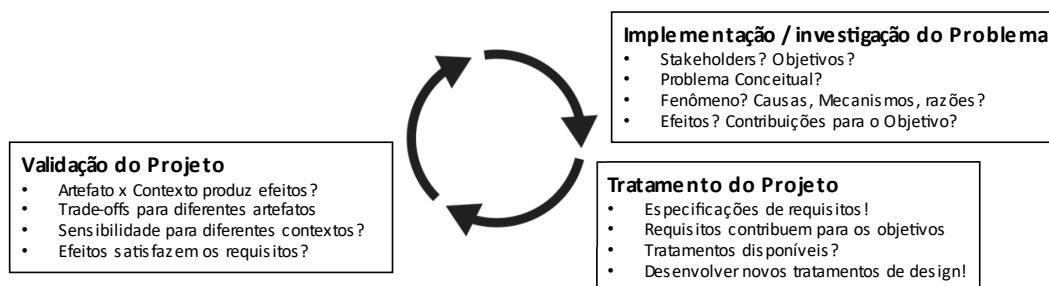


Figura 3.2: Ciclos de engenharia.

Fonte: adaptado de Wieringa (2014).

Estes ciclos de pesquisa começam com a “investigação do problema” uma fase considerada como uma questão acerca do conhecimento. Sua parte teórica fica bastante evidenciada pela busca de informação como um modo de entender o problema, sem ter ainda a capacidade de transformá-lo. A etapa seguinte, o “tratamento do projeto” (Solução de Design), é caracterizada pelo enfrentamento de um problema prático, que pode ter como etapa a decomposição dos problemas e especificações de requisitos e suas soluções. A etapa de validação do projeto é justificar que ele contribua para os objetivos das partes interessadas(stakeholders). Para Wieringa (2014) no ciclo de engenharia, a validação é feita antes da implementação, que consiste em investigar os efeitos da interação entre um protótipo de um artefato e um modelo do contexto do problema e de compará-los com exigências do tratamento.

### 3.1.2 As Partes Interessadas (Stakeholders) e o Contexto Social

Os projetos de DSR são problemas de melhoria. Cada um dos problemas tem um contexto no qual se pretende alguma melhora, e para entender o problema de DSR, esse contexto tem que ser entendido. Para Wieringa (2014), o contexto do problema de um artefato pode ser estendido com as partes interessadas do artefato e com o conhecimento usado para projetar o artefato.

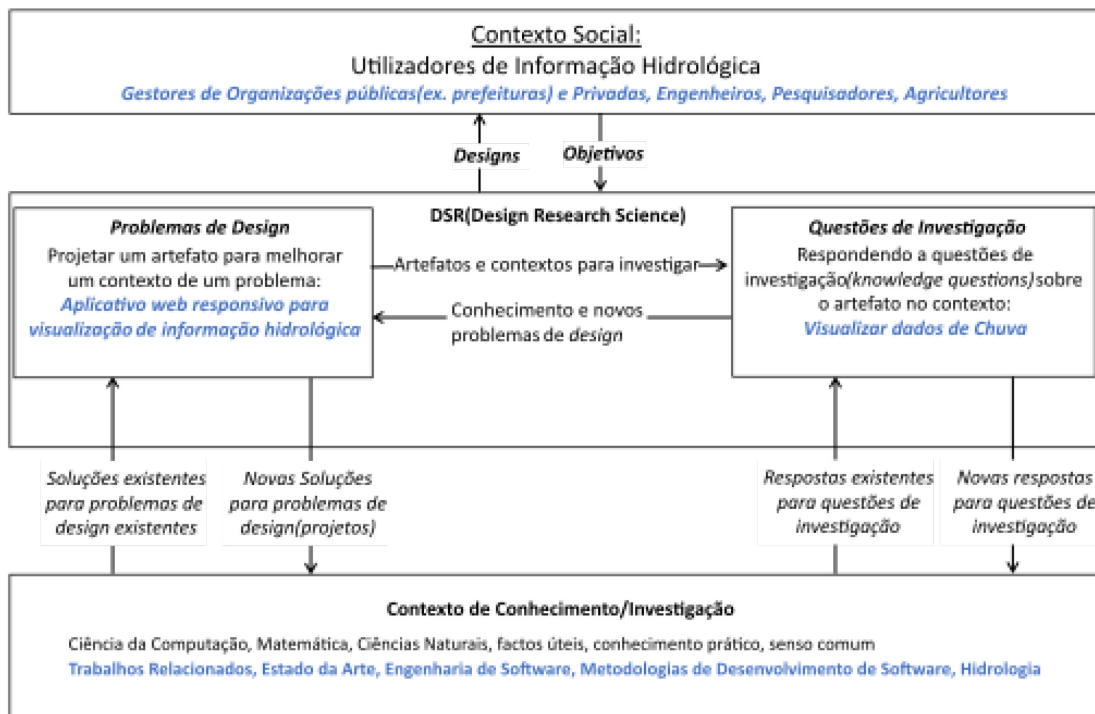


Figura 3.3: Contexto do projeto de ciência do design.

Fonte: adaptado de Wieringa (2014).

O contexto social contém as partes interessadas que podem afetar o projeto ou podem ser afetadas por ele. As partes interessadas incluem possíveis usuários, operadores, mantenedores. O contexto do conhecimento consiste em teorias existentes da ciência e da engenharia, especificações de projetos atualmente conhecidos, fatos úteis sobre produtos atualmente disponíveis, lições aprendidas com a experiência de pesquisadores em projetos de ciência do design e bom senso. Este contexto estendido é o contexto do projeto de ciência do design como um todo.

Na Figura 3.3, também têm-se as interações de problema de design, que é um problema para reprojeter um artefato para que ele contribua melhor para a realização de algum objetivo. Os problemas de design assumem um contexto e objetivos das partes interessadas e exigem um artefato de modo que as interações de (contexto do artefato) ajudem as partes interessadas a atingir seus objetivos em DSR, especificamos requisitos para o artefato que são motivados pelos objetivos das partes interessadas. Isso nos dá o esquema ou modelo para expressar problemas de projeto (Wieringa, 2014).

Tabela 3.3: Esquema para problemas de design

- |  |
|--|
| <ul style="list-style-type: none"><li>• <b>Melhorar</b> &lt;contexto de um problema&gt;</li><li>• <b>Modo</b> &lt;(re)projetando um artefato&gt;</li><li>• <b>Que satisfaz</b> &lt;alguns requisitos&gt;</li><li>• <b>Em ordem para</b> &lt;ajudar as partes interessadas a alcançar objetivos&gt;</li></ul> |
|--|

Fonte: adaptado de Wieringa (2014).

Segundo *Wieringa* (2014), nem todos os elementos do modelo de problema de design podem ser conhecidos no início do projeto, e alguns podem ser inventados como parte de um cenário de aplicação hipotético. Declarar seu problema de design de acordo com um esquema é útil porque ajuda você identificar informações faltantes que são necessárias para limitar sua pesquisa problema.

Com o entendimento da Tabela 3.3, pode-se aplicá-la no presente projeto segundo a proposição que consta da Tabela 3.4.

Tabela 3.4: Esquema de design aplicado ao projeto de aplicativo

- |   |
|---|
| <ul style="list-style-type: none"><li>• <b>Melhorar:</b> o acesso e visualização de dados hidrológicos</li><li>• <b>Modo:</b> Projetar um aplicativo web</li><li>• <b>Que satisfazem:</b> os requisitos de processamento, consistência, facilidade, rapidez, responsividade.</li><li>• <b>Em ordem para:</b> ajudar os técnicos, engenheiros hidrólogistas, pesquisadores, meteorologistas e gestores a tomar decisões relacionadas a seus trabalhos.</li></ul> |
|---|

Fonte: adaptado de Wieringa (2014).

### 3.2 Seleção de técnicas, abordagens e processos de Engenharia de Software

Processos de software reais são sequências intercaladas de processos técnicos, colaborativos e atividades gerenciais com o objetivo geral de especificar, projetar, implementar, e testar um sistema de software (Sommerville, 2016).

Roger e Bruce (2020) identificam um pequeno número de atividades-quadro que são aplicáveis a todos os projetos de software, independentemente de seu tamanho ou complexidade. No contexto da engenharia de software, um processo não é uma prescrição rígida de como construir software de computador. Em vez disso, é uma abordagem adaptável que permite que as pessoas que fazem o trabalho (a equipe de software) escolham o conjunto

apropriado de ações e tarefas de trabalho. Uma estrutura de processo estabelece a base para um processo completo de engenharia de software, identificando um pequeno número de atividades da estrutura que são aplicáveis a todos os projetos de software, independentemente de seu tamanho ou complexidade. Além disso, a estrutura do processo engloba um conjunto de atividades abrangentes que são aplicáveis em todo o processo de software. Esta estrutura de processo genérica para engenharia de software englobaria cinco atividades:

Tabela 3.5: Estrutura das atividades do processo de engenharia de software.

<b>Comunicação</b>	A intenção é entender os objetivos das partes interessadas para o projeto e reunir os requisitos que ajudam a definir os recursos e funções do software.
<b>Planejamento</b>	Define o trabalho de engenharia de software descrevendo as tarefas técnicas a serem executadas.
<b>Modelagem</b>	Criação de modelos para entender melhor os requisitos de software e a projeto que atenderá a esses requisitos.
<b>Construção</b>	Atividade combina a geração de código os testes que são necessários para descobrir erros no código.
<b>Implantação</b>	O software (como uma entidade completa ou como um incremento parcialmente concluído) é entregue ao cliente que avalia o produto entregue e fornece feedback com base na avaliação.

Fonte: adaptado de Roger & Bruce (2020).

Já para Somerville (2016), as quatro atividades básicas de um processo de engenharia de software seriam as seguintes: especificação, desenvolvimento, validação e evolução são organizadas de forma diferente em diferentes processos de desenvolvimento. Esses processos podem ser aplicados de diferentes maneiras, por exemplo no modelo em cascata (*Waterfall*), eles são organizados em sequência, enquanto no desenvolvimento incremental eles são intercalados. Como essas atividades são realizadas depende do tipo de software que está sendo desenvolvido, a experiência e competência dos desenvolvedores e a tipo de organização que desenvolve o software.

Da mesma forma, as metodologias de engenharia de software não prescrevem uma sequência rígida de atividades a depender da metodologia que for escolhida. Os ciclos de engenharia apresentados na metodologia de DSR (*Design Research Search*) não informam como gerenciar os processos, podendo-se escolher o processo que melhor se adapta à especificidade do projeto. Neste projeto de dissertação, escolheu-se a metodologia de prototipagem como processo de desenvolvimento.

Segundo Roger e Bruce (2015), frequentemente, as partes interessadas definem os objetivos gerais de um software, mas não identificam requisitos detalhados para as funções

e recursos. Para os autores, o desenvolvedor pode não ter certeza da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação homem-máquina deve tomar. Dentre essas e em muitas outras situações, um paradigma de prototipagem pode oferecer a melhor abordagem.

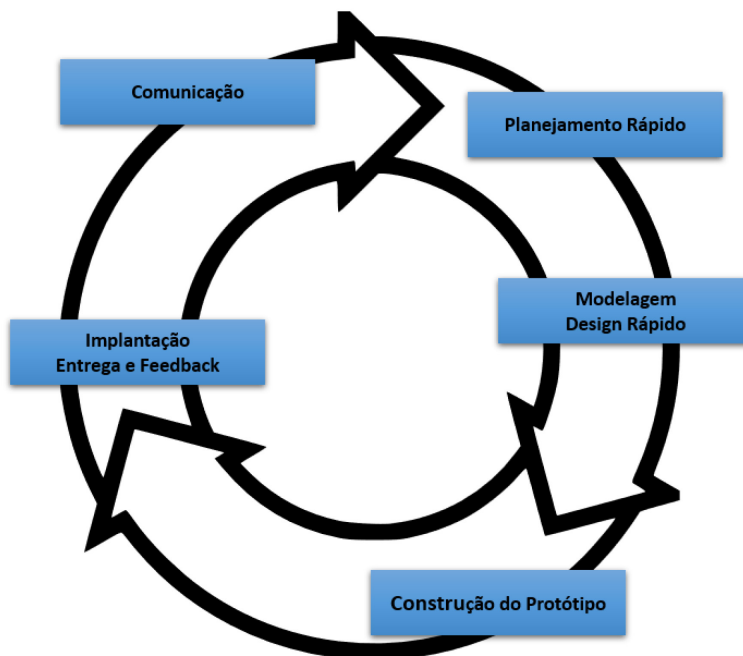


Figura 3.4: Paradigma de prototipagem.  
Fonte: adaptado de Roger e Bruce (2015)

O paradigma da prototipagem (Figura 3.4) começa com a comunicação. Você se encontra com as partes interessadas para definir os objetivos gerais para o software, identificar os requisitos que são conhecidos e delineiam as áreas onde uma definição mais detalhada é obrigatória. Uma iteração de prototipagem é planejada rapidamente e a modelagem (na forma de um projeto”) ocorre. Um design rápido concentra-se em uma representação desses aspectos do software que será visível para os usuários finais (por exemplo, layout de interface humana). O design rápido leva à construção de um protótipo. O protótipo é implantado e avaliado pelas partes interessadas, que fornecem feedback que é usado para refinar ainda mais os requisitos. A iteração ocorre à medida que o protótipo é ajustado para satisfazer a necessidades de várias partes interessadas, ao mesmo tempo em que permite que você compreenda melhor suportar o que precisa ser feito. Idealmente, o protótipo serve como um mecanismo para identificar os requisitos de software. Se um protótipo funcional deve ser construído, você pode usar fragmentos de programa existentes ou aplicar ferramentas que permitem a geração rápida funcional de programas.

Independentemente da forma como é aplicado, o paradigma de prototipagem ajuda você e outras partes interessadas a entender melhor entender o que deve ser construído quando os requisitos não estão claros (Roger & Bruce,2015).

### 3.2.1 *Requisitos do software*

A especificação de software ou engenharia de requisitos é o processo de compreensão e definição de quais serviços são necessários do sistema e identificar as restrições sobre a operação e desenvolvimento do sistema (Sommerville, 2016).

Para Bruce & Roger (2015), engenharia de requisitos é o termo para o amplo espectro de tarefas e técnicas que levam ao entendimento dos requisitos. Do ponto de vista do processo de software, a engenharia de requisitos é uma importante ação de engenharia de software que começa durante a atividade de comunicação e continua na atividade de modelagem.

Segundo Sommerville (2016), deve ser adaptado às necessidades do processo, do projeto, do produto e das pessoas que fazem o trabalho, notando-se que cada uma dessas tarefas é feita de forma iterativa à medida que a equipe do projeto e as partes interessadas continuam a compartilhar informações sobre suas respectivas preocupações.

### 3.2.2 *Técnicas de elicitação de requisitos*

A elicitação de requisitos envolve reuniões com as partes interessadas de diferentes tipos para descobrir informações sobre o sistema proposto (Sommerville, 2016). Complementa-se esta informação com conhecimento de sistemas existentes e seu uso e informações de documentos de vários tipos. É importante entender como as pessoas trabalham, o que eles produzem, como eles usam outros sistemas e como eles podem precisar mudar para acomodar um novo sistema.

Para entender como as pessoas trabalham, existem duas abordagens fundamentais para a elicitação de requisitos:

- entrevistas em que se pergunta as partes interessadas sobre o que elas fazem;
- observação ou etnografia, onde você observa as pessoas fazendo seu trabalho para ver quais artefatos eles usam, como eles os usam e assim por diante.

### 3.2.3 *Identificação das fontes dos requisitos*

Para este trabalho, utilizaram-se as seguintes técnicas para elaboração dos requisitos: (i) entrevistas semiestruturadas com gestores, engenheiros, meteorologistas e pesquisadores ligados a áreas ambientais e de hidrologia; (ii) pesquisa relacionada documentação existente sobre dados hidrológicos e ambientais, dados de satélite sobre precipitação, repositórios de dados, organizações ligadas a estes dados; (iii) levantamento relacionado com os repositórios e ferramentas existentes que disponibilizam informação hidrológica na Internet.

### 3.2.4 *Modelagem do sistema e dos requerimentos*

A modelagem de requisitos usa uma combinação de texto e formas esquemáticas para descrever os requisitos de uma maneira relativamente fácil de entender e, mais importante, simples de revisar para correção, integridade e consistência (Roger & Bruce, 2015). Para Sommerville (2016), a modelagem de sistemas é o processo de desenvolvimento de modelos abstratos de um sistema, com cada modelo apresentando uma visão ou perspectiva diferente desse sistema. Para Roger e Bruce (2020), os modelos de requisitos podem ser prontamente avaliados por todas as partes interessadas, resultando em *feedback* útil o mais cedo possível. Mais tarde, à medida que o modelo é refinado, torna-se a base para o projeto de software.

### 3.2.5 *Projeto de arquitetura*

Segundo Roger & Bruce (2020), o projeto de arquitetura representa a estrutura de dados e componentes de programa que são necessários para construir um sistema baseado em computador. Ele considera o estilo arquitetônico que o sistema tomará, a estrutura e as propriedades dos componentes que constituem o sistema e as inter-relações que ocorrem entre todos os componentes de um sistema.

Para Sommerville (2016), o projeto de arquitetura está preocupado em entender como um sistema de software deve ser organizado e projetar a estrutura geral desse sistema. O projeto de arquitetura é a primeira etapa do processo de projeto de software. É o elo crítico entre design e engenharia de requisitos, pois identifica os principais componentes estruturais em um sistema e as relações entre eles. A saída do processo de projeto arquitetônico é um modelo de arquitetura que descreve como o sistema é organizado como um conjunto de componentes que se comunicam entre si.

### 3.2.6 *Implementação de componentes*

O projeto e implementação de software é o estágio do processo de engenharia de software no qual um sistema de software executável é desenvolvido. Para alguns sistemas simples, engenharia de software significa projeto e implementação de software e todas as outras atividades de engenharia de software são mescladas com esse processo. No entanto, para grandes sistemas, projeto e implementação de software é apenas um dos vários processos de engenharia de software (engenharia de requisitos, verificação e validação etc.) (Sommerville,2016).

### 3.2.7 *Teste de software*

Nesta etapa, o software é testado para descobrir erros que foram cometidos inadvertidamente à medida que foi projetado e construído. Uma estratégia de teste de componentes de software considera testar componentes individuais e integrá-los em um sistema como um todo. O teste destina-se a mostrar que um programa faz o que se destina a fazer e descobrir defeitos do programa antes de ser colocado em uso.

## Requisitos e Projeto de Arquitetura

Neste capítulo, uma vez apresentada a metodologia (Capítulo 3) concebida para o trabalho, especifica-se o desenvolvimento do projeto. Utilizando técnicas de engenharia de software e experiência de usuário, inicia-se o desenvolvimento do projeto do protótipo, utilizando. Na seção 4.1, são descritos excertos das entrevistas, com potenciais usuários e partes interessadas. Na seção 4.2, são elicitados os requisitos, sendo que este processo divide-se em dois passos: definição de requisitos funcionais e definição de requisitos não funcionais.

Após a especificação dos requisitos, será descrita a escolha dos componentes (Seção 4.3) e tecnologias baseados na pesquisa e consulta do estado da arte baseado no capítulo dois. Esta seção divide-se em dois passos: (i) escolha de tecnologias de interface de usuário (*frontend*<sup>86</sup>) (Seção 4.3.1) e (ii) escolha de tecnologias que vão compor a camada de suporte e a camada de aplicação do lado do servidor (*backend*<sup>87</sup>) (Seção 4.3.2).

Com as escolhas das tecnologias e ferramentas, define-se o projeto de arquitetura<sup>88</sup> (Seção 4.3) que atenderá os requisitos funcionais e não funcionais de um sistema.

---

<sup>86</sup> *Frontend* é uma parte fundamental do desenvolvimento web e de aplicativos que se concentra na criação da interface do usuário (UI) e na interação do usuário com um site ou aplicativo. É a camada visível de um sistema de software com a qual os usuários interagem diretamente. O desenvolvedor frontend é responsável por projetar, desenvolver e otimizar a parte visual e funcional de um site ou aplicativo, tornando-o acessível e agradável para os usuários.

<sup>87</sup> O *backend* é a parte não visível de um sistema de software, que é responsável por processar os dados, a lógica de negócios e a comunicação com o servidor. Ele é o cérebro por trás de um site, aplicativo ou serviço web e lida com as funcionalidades que não são diretamente visíveis aos usuários finais. O *backend* é uma parte essencial do desenvolvimento web e de aplicativos e desempenha várias funções críticas

<sup>88</sup> Um projeto de arquitetura de software é um plano ou estratégia abrangente que descreve a estrutura, componentes, relacionamentos e design de um sistema de software. Ele é um estágio crítico no ciclo de desenvolvimento de software e é responsável por definir as diretrizes que guiarão o desenvolvimento, a implementação e a manutenção do software. O objetivo principal de um projeto de arquitetura de software é criar uma base sólida e escalável para o sistema de software, garantindo que ele atenda aos requisitos funcionais e não funcionais, como desempenho, segurança, escalabilidade e usabilidade.

## 4.1 Entrevistas

Para a eliciação de requisitos, foram conduzidas entrevistas abertas por meio de videoconferência online, com especialistas da área: pesquisadores, engenheiros e meteorologistas.

Seguem-se excerto de quatro trechos de entrevistas. O formato da entrevista foi aberto. Nesse tipo de entrevista, o entrevistador não segue um roteiro fixo de perguntas. Em vez disso, permite que a conversa flua de maneira mais livre, permitindo que o entrevistado expresse suas ideias e perspectivas sem restrições significativas.

Todos os entrevistados foram perguntados como acessam e lidam com dados hidrológicos no seu dia a dia. A primeira é uma entrevista com a pesquisadora L. dos Santos, observando-se que ela trabalha com dados hidrológicos. O segundo excerto é de R. Costa, um engenheiro que trabalha com projetos de engenharia correlatos a hidrologia. O terceiro excerto de entrevista selecionado tem como perfil uma meteorologista, C. Avelar. O quarto excerto é uma entrevista com a engenheira L. Araújo, que trabalha na Agência Nacional de Águas e Saneamento Básico (ANA).

### Excerto nº 1:

*“Estou trabalhando na bacia do São João com produtos de sensoriamento remoto para produzir estimativas de variáveis hidrológicas de precipitação e evapotranspiração. Sendo que está tendo problemas na bacia porque é pouco monitorada e não tem muitos dados de vazão. E para modelagem hidrológica é muito ruim, tem que usar de regionalização para tentar extrair informação, tem poucas estações de precipitação, então o sensoriamento remoto é muito útil por isso....*

...

*Na parte de peneirar os dados, é difícil devido a poucas fontes, e nem sempre o que se tem disponível ajuda. As estações que existem têm falhas; no caso de uma bacia que tem um monitoramento precário é pior ainda, porque você não consegue preencher essas falhas, são necessárias outras bacias próximas para preencher dados de outra...*

....

*Assim eu utilizo linguagem de programação python e softwares proprietários que pegam os dados do Hidroweb, e consegue-se montar os gráficos de vazão, mas não me adaptei aos softwares e prefiro utilizar o Excel para gerar os gráficos. Porque quando os dados vêm do Hidroweb, não vem de um jeito muito útil. Os dados vêm na horizontal e é difícil e trabalhoso*

*no Excel. Mesmo no Excel ainda a curva de aprendizagem não é rápida.”*

**Excerto nº 2:**

*“Sou engenheiro civil, de uma empresa privada de engenharia, e sou responsável pelo setor de hidrologia e drenagem, A empresa faz projetos, não lida com as obras em si... Nesses projetos de hidrologia e drenagem usamos corriqueiramente os dados hidrológicos pluviométricos e fluviométricos. Os dados que utilizamos são oriundos das agências brasileiras, principalmente o Hidroweb da ANA (Agência Nacional de Águas e Saneamento). No entanto utilizamos outros sites que ofertam estes dados como CEMADEN, INEA, e ALERTA RIO no caso de obras no município do Rio de Janeiro, INMET quando precisamos de um dado infra diário de estação gerenciada pelo INMET e também usamos outros produtos de escala global de agências internacionais como NASA, ESA, JAXA, CMSA etc. tanto de georreferenciamento quanto de precipitação de radar e outros sensores, principalmente em áreas remotas que tenhamos alguma dúvida em relação as séries que são disponibilizadas pelas agências nacionais...*

*Fazemos uma validação, tentamos completar esses dados, em lugares onde temos mais dúvidas. Nós acabamos fazendo essa validação com esses dados espaciais...*

*Esses usos desses dados climatológicos principalmente pluviométricos e fluviométricos, nós usamos para alimentar e subsidiar os projetos de infraestrutura: Bueiros, pontes, segurança nas estradas, projetos rodoviários, ferroviários, aeroportuários. Nos projetos de estrutura de drenagem necessitamos sempre desde dados pluviométricos e fluviométricos atualizados.... Sendo que estes projetos podem ser de requalificação fluvial...*

*Por exemplo, agora, teve uma tragédia na cidade de Petrópolis, têm que ser reconstruídos rios, canais e obras de drenagem. Vamos precisar pegar estes dados incluindo as atualizações com estes eventos extremos, refazer toda uma estatística, para ter as equações de chuva intensa que vão subsidiar os projetos de todas as estruturas de drenagem...*

*Nós precisamos mais dos eventos extremos e das máximas. Séries históricas mais longas possíveis. Encontramos dificuldades nos sites de ter séries históricas longas, arquivos pesados, acesso dificultado....*

*Normalmente só baixa de ano em ano, e as vezes quando é um dado infra diário o intervalo para baixar é menor ainda. Isto dificulta a aquisição destes dados, para montarmos toda uma série efetuar um processamento de acordo com os algoritmos que temos na empresa....*

*As plataformas disponíveis não permitem uma fácil visualização da extensão histórica dos dados, nos próprios campos das plataformas das agências estão em atributos diferentes as*

*séries de início e final e não raro, estão completados erroneamente, com dados que não compreendem a realidade....*

*Então, ali, nos atributos do posto, deveriam estar o número de anos que você tem de registro, o início, o final, anos com falha, algum resumo mais apropriado e transparente daquele registro de dados e isso não acontece... Por exemplo, ano inicial não bate com a realidade, os anos consistidos não constam nos atributos... Tem que abrir o arquivo e pesquisar. São informações que são muito necessárias para nós. Saber quantos anos completos, quantos anos não completos, quantos anos consistidos, quantos anos não consistidos. Se essa análise já está feita, é muito melhor para gente.”*

**Excerto nº 3:**

*“Eu sou meteorologista, trabalho no INEA desde 2014, a minha área de atuação é o monitoramento hidrometeorológico com foco em alerta de cheia. Para fazer essa tomada de decisão para envio de alertas, precisamos de uma rede de monitoramento, não só as estações de superfície, que é a informação que nós obtemos usualmente pontualmente de chuva em nível de rio, mas também é necessário ter acesso a dados de radar, de satélite, de detecção de descargas atmosféricas, entre outros, para que possamos antecipar este envio de alerta de cheias de maneira que a meteorologia forneça para nós o que esperar em relação a hidrologia, que é o foco do INEA como órgão ambiental responsável por executar as políticas ambientais de recursos hídricos. O INEA identificou esta necessidade por monitoramento meteorológico para essa aplicação em hidrologia lá atrás desde o desastre de 2011 na região serrana do Rio de Janeiro; em 2012, foi feito um processo de aquisição, e, em 2014, foram instalados 2 radares meteorológicos. ...*

*Esses radares são superimportantes para que consigamos, da melhor maneira possível, antecipar o envio dos alertas, visto que nossas bacias hidrográficas, boa parte delas, estão inseridas em vales encaixados; então, o tempo de respostas das chuvas para a subida do nível do rio é muito curto, e o monitoramento vai adicionar; o que é feito em grandes bacias, que se baseia apenas na hidrologia, não é suficiente para antecipar os eventos para a defesa civil. É necessário um monitoramento meteorológico antevendo-se à resposta dos níveis dos rios...*

*Utilizamos um software da fabricante do radar, que é capaz de fazer o pós-processamento dos dados e gerar os produtos e converter as imagens para o formato que for necessário, tanto para atendimento interno, quanto para compartilhamento das informações com as outras instituições com as quais mantemos acordo de cooperação técnica. E nós ainda não*

*conseguimos desenvolver ou contratar uma ferramenta que faça a integração desses dados de diferentes fontes. Então, hoje, a gente analisa os dados dos radares no nosso software do radar e analisamos os dados das outras fontes nas suas plataformas próprias. Então, além dos radares, nós temos uma rede de estações de monitoramento de superfície, que são estações hidrometeorológicas. Elas monitoram chuvas e nível de rio. Hoje, contamos com 104 estações deste tipo. E essas estações têm os dados enviados para o servidor do INEA, e esse servidor hospeda um sistema que disponibiliza esses dados para o público. É um outro tipo de informação que temos, que, neste momento, não está integrado com os dados de radar. Essas informações são analisadas separadamente, mas todas elas colaboram para a tomada de decisão e o envio de alerta. Além dessa rede hidrometeorológica e dos radares meteorológicos, nós temos uma contratação de um serviço de dados de descargas atmosféricas. Esse serviço é de uma outra fornecedora; esta fornecedora tem uma rede de monitoramento própria de pulsos eletromagnéticos propagados na atmosfera, e eles tem o seu próprio software, que acessamos via web. Uma página com login e senha que, por contrato, tem acesso restrito, já que é um serviço pago. Nesses acessos, nós temos todos os flashes, os pulsos, uma classificação da descarga para saber se ela entra na nuvem ou no solo, a quantidade de descargas que estão sendo identificadas e análises estatísticas de identificação de célula de tempestade, uma indicação para onde ela está se deslocando. Ele dá um bom indicativo do grau de severidade das tempestades. Normalmente, quanto mais severas, nós temos uma taxa de descarga atmosféricas um pouco maior....*

*Nós temos um protocolo de envio de alerta que inclui contatos com as defesas civis municipais, especialmente dos municípios sobre os quais nós mantemos monitoramento hidrometeorológico. Não são todos, temos uma rede de estações. Essa rede de estações de superfície é a nossa base para o atendimento dos municípios, visto que os nossos alertas, são alertas de cheias. Então, precisamos da informação do nível do rio para validar os alertas. Onde temos o equipamento instalado para monitoramento do rio, esse equipamento serve para fins de alerta, porque esse dado pode ter outras finalidades também, que não só o alerta, como monitoramento básico, para entender a água no sistema para abastecimento, para segurança hídrica e para dimensionamento de projeto como barragens. Os equipamentos, que estão instalados em áreas urbanas que você tem uma ocupação ribeirinha e que já têm um histórico de cheias na região, são utilizados para envio de alerta. E esses municípios atingidos estão cadastrados no sistema. Hoje, nós fazemos a comunicação, por WhatsApp, a forma mais rápida que temos, por uma lista de transmissão de municípios cadastrados, que tentamos atualizar de tempos em tempos, porque esta*

*atuação em defesa civil municipais é muito dinâmica; entra governo e sai governo e mudam os agentes e precisamos manter este cadastro atualizado. Também outros componentes do sistema, como a defesa civil estadual, atuam. Temos a nossa página que reflete os alertas que são enviados e tem acesso público. As imagens do radar são atualizadas de 5 em 5 minutos... Nós também analisamos dados de precipitação para secas mensalmente...”*

**Excerto nº 4:**

*“Eu trabalho, hoje, na agência nacional de águas há 11 anos, mas, antes, eu trabalhei 17 anos no Serviço Geológico do Brasil, e, antes, trabalhei 10 anos como consultora na iniciativa privada. Sempre trabalhei com dados hidrológicos na maior parte do tempo e, hoje em dia, trabalho com dados de saneamento básico e dados de segurança de barragem....*

*Com relação a hidrologia, a ANA tem uma atribuição importante em gerenciar o banco de dados da rede hidro hidrometeorológica nacional com vistas a promover a operação da rede e, ao mesmo tempo, guardar e disponibilizar os dados.*

*Tem também uma área de estudos, que inclui o sistema nacional de informações sobre os recursos hídricos, que também é uma outra atribuição da ANA, com o dado trabalhado para tomada de decisão. Consolidamos muitos mapas de geociências, geografia e socioeconômicos com dados abertos com cruzamentos de informações com recursos hídricos, produzindo mapas que são muito utilizados pelo Brasil inteiro. É um apoio ao gerenciamento de recursos hídricos....*

*No caso dos dados hidrológicos, em especial, o Hidroweb que a ANA tem a atribuição de armazenar, alimentar e tem capacidade para receber dados de todas as estações hidrometeorológicas do Brasil. A maior parte delas está cadastrada lá, porém algumas entidades não cadastram os dados lá, então você não tem acesso a existência da estação e quando tem muitas vezes você não tem acesso aos dados daquelas estações cadastradas, pois esta adição dos dados é voluntária. Ela não é obrigatória. Existe uma resolução conjunta da ANA e ANEL que faz com que os dados sejam depositados obrigatoriamente pelos concessionários de energia de estações que eles são obrigados a instalar ao redor do aproveitamento para montante e jusante e mandar estes dados para base de dados. Estes dados são dados telemétricos e com intervalo de mediação menor de 1 dia. Outras entidades têm suas estações cadastradas como por exemplo o CEMADEN, mas os dados não estão lá, com formato bem diferente. Então assim essa questão dos dados, é importante também ver que os formatos não são exatamente os mesmos, não só na discretização, no tempo e a forma*

*de armazenar cada um tem a sua. É importante que exista uma iniciativa de buscar esse dado e colocar num formato que seja importante para o usuário, que ele possa transformar esse formato de uma maneira mais imediata, mais fácil, para utilizar mais rapidamente o dado. O CEMADEN por exemplo tem dados de muitas estações mais recentes, e não tem um histórico muito longo, por que foi uma rede criada a partir dos grandes desastres de 2011 em diante, e as estações foram instaladas mais recentemente...”*

Com os excertos das entrevistas realizadas, apresenta-se algumas considerações entre as quais destaca-se pontos para ajuda na elicitação dos requisitos, pois caracterizaram a dificuldades dos usuários no acesso e processamento para criação dos produtos a serem utilizados em diversas aplicações. Por software proprietário e de código aberto não terem uma interface amigável com uma boa experiência de usuário optam muitas vezes pela utilização de software já conhecidos por eles, com uma curva menor de aprendizado e agilidade maior, por exemplo com software de criação de planilhas como Excel.

A falta de monitoramento em regiões esparsas no Brasil, a inconsistência, a falha e falta de dados, a falta de registros de início e fim das estações para facilitar a seleção inicial acarreta perda de tempo e dificuldade na seleção das estações.

Um dos entrevistados relatou a necessidade do acesso e identificação dos eventos extremos e máximas para subsidiar os projetos de estruturas. Também relatou a dificuldade em baixar dados para períodos mais longos para dados de pequenos intervalos, para montar toda uma séria efetuar o processamento com seus algorítmicos preferidos.

Os entrevistados apontaram que não disponibilizam uma fácil visualização da extensão histórica (período total), sendo necessário buscar dentro do arquivo tais atributos. Outra entrevistada mencionou a falta de integração nos próprios sistemas dentro de sua organização e formatação diferente entre entidades.

Por último, as entrevistas permitiram entender o contexto em que o software será usado e perceber as necessidades dos usuários para elicitação dos requisitos na próxima seção do capítulo. Estas entrevistas serão utilizadas para a criação de *Personas* no Capítulo 6. As *personas* são personagens fictícios que representam segmentos de usuários reais e são construídas com base em dados demográficos, comportamentais, necessidades, motivações e características específicas.

## 4.2 Elicitação de Requisitos

Para a análise dos requisitos, utilizaram-se entrevistas com diferentes profissionais das áreas de meteorologia/hidrologia/climatologia e ciências aplicadas, como citadas na seção 4.1, bem como diversas conversas com os orientadores ao longo do desenvolvimento desta dissertação para gerar uma lista de ideias, de componentes e um refinamento de funcionalidades.

Os requisitos funcionais constituem as operações que o usuário do sistema deve ser capaz de fazer, e requisitos não funcionais são tudo o mais sobre o sistema, exceto a funcionalidade, como, por exemplo, a usabilidade da aplicação.

### 4.2.1 Requisitos Funcionais

- RF#-1 - O usuário visualiza estações pluviométricas/fluviométricas cadastradas no SNRH espacialmente distribuídas em um mapa geográfico.
- RF#-2 - O usuário visualiza os parâmetros básicos das estações no mapa.
- RF#-3 - O usuário visualiza médias mensais de chuvas por ano in situ das estações
- RF#-4 - O usuário visualiza médias mensais de chuva por satélite das estações
- RF#-5 - O usuário visualiza médias diárias de chuva in situ das estações
- RF#-6 - O usuário visualiza médias diárias de chuva por satélite das estações
- RF#-7 – O usuário visualiza data e valor da máxima diário dentro de um mês para dados in situ.
- RF#-8 – O usuário visualiza data e valor da máxima diário dentro de um mês para dados de satélite.
- RF#-9 - O usuário poderá comparar chuvas in situ com dados processados por satélite por ano.
- RF#-10 - O usuário poderá descarregar os dados de precipitação in situ.
- RF#-11 - O usuário poderá descarregar os dados de precipitação por satélite.1
- RF#-12 - O usuário poderá visualizar um mapa de densidade de calor de precipitação da América do Sul dos últimos dias.
- RF#-13 - O usuário pode buscar por nome da estação
- RF#-14 – O usuário pode visualizar data de início de série histórica

Os casos de usos derivados e selecionados das entrevistas podem ser visualizados no diagrama da Figura 4.1, e são uma forma de descrever as interações entre os usuários e um

sistema usando um modelo gráfico e texto estruturado.

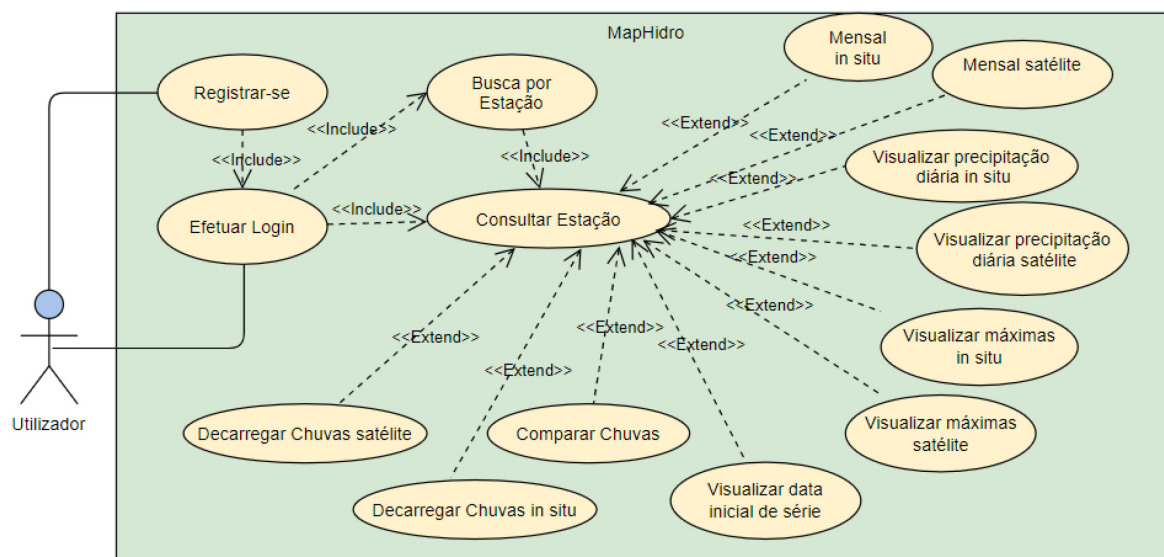


Figura 4.1: Diagrama de caso de uso da aplicação web MapHidro.

#### 4.2.2 Requisitos não funcionais

A lista de requisitos não funcionais, excluindo a escolha da linguagem de programação, plataforma, é a seguinte:

- acessibilidade por URL – a aplicação web é acessível através de maioria dos navegadores Internet em URL para todos com acesso à Internet.
- internacionalização – a aplicação web está disponível em utilização em vários idiomas.
- usabilidade – a aplicação permite visualização das funcionalidades em diferentes equipamentos (smartphones, tablets, desktop, notebooks)

### 4.3 Escolha de Componentes

No desenvolvimento de software de hoje, existem várias maneiras de projetar e implementar uma aplicação web, com opções de escolha de ferramentas para facilitar a implementação, desde a escolha do tipo de aplicação, seguida pela escolha do ambiente de desenvolvimento com as chamadas IDEs (Ambiente de Desenvolvimento Integrado) que reúnem características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo, tecnologias e *frameworks* que compreendem um conjunto de classes implementadas em uma linguagem de programação específica, usadas para auxiliar o desenvolvimento de software.

A primeira escolha que necessita ser feita é que tipo de aplicação é melhor para visualização dos gráficos e mapas. Considerando o domínio da aplicação web, potenciais usuários e casos de uso do sistema, existem duas abordagens principais que seriam possíveis, implementar software com tecnologia nativa tanto para computadores, quanto para smartphones ou criar uma aplicação web que utilizariam os navegadores dos dispositivos..Após fazer a escolha do tipo de aplicação, deve-se reconhecer que os principais tipos de dispositivos têm diferentes sistemas operacionais e características. Os usuários de dispositivos móveis podem ser relutantes a instalar um novo software. O mesmo pode ser dito com novos programas para desktop. Instalar um programa de desktop pode parecer uma espécie de compromisso para um usuário. Além disso, desenvolver uma única aplicação web com design responsivo demanda muito menos tempo do que criar um aplicativo nativo para cada sistema operacional de ambiente móvel e desktop (Windows, Mac, Linux). É claro que as aplicações web têm suas desvantagens – é difícil garantir uma boa experiência do usuário para usuários de todos os navegadores e dispositivos da Internet de todos os tamanhos diferentes, e eles tendem a ser mais lentos e menos ajustados do que seus equivalentes nativos. No entanto, nesta situação, as vantagens parecem superar as desvantagens, pelo que se decidiu desenvolver uma aplicação web.

Ao escolher o framework (ambiente de produção<sup>89</sup>) e a linguagem de programação, vários fatores devem ser levados em consideração – em primeiro lugar, a decisão sobre o tipo de aplicação deve ser refletida – o que torna possível utilizar substancialmente menos componentes ou tecnologias menos complexas em termos de implementação.

Outro aspecto que vale a pena levar em consideração é a popularidade de uma linguagem e ambiente/estrutura de produção, pois as linguagens de programação mais populares têm comunidades maiores e, portanto, é muito mais fácil encontrar soluções para problemas, que sem dúvida, ocorrerão.

O elemento final e com maior peso foi usar tecnologias que permitissem um melhor desempenho nos dispositivos móveis. Tecnologias recentes da Web, como HTML5, JavaScript e WebGL, permitiram aplicações de mapas web altamente dinâmicos executados em navegadores da Web padrão. A complexidade para o desenvolvimento de tais aplicações web foi bastante reduzida por bibliotecas de mapas web em conjunção com ambientes de produção modernos.

---

<sup>89</sup> tradução nossa

Os resultados experimentais de Zunino et al. (2020), concluíram que o Mapbox/MaplibreGLJS tem o melhor desempenho geral em dispositivos de gama média e alta em aplicações de mapa web em relação às bibliotecas Openlayers e Leaflet.

Em relação a tecnologias utilizadas no *Backend*<sup>90</sup>, experimentos com tecnologias de servidores web demonstraram que Node.js é mais eficiente em relação a tecnologias como Python-Web e PHP (Lei et al., 2014) e tecnologias .NET que estão hospedados nos Serviços de Informações da Internet (Servidor IIS). (Chitra, & Satapathy, 2017).

Com respeito às tecnologias de servidores web há Apache Web Server<sup>91</sup> e Nginx<sup>92</sup> de código aberto. Os resultados experimentais mostram que o Nginx oferece melhor desempenho em termos de capacidade de resposta e escalabilidade, enquanto sua eficiência degrada em cargas pesadas. O servidor web Nginx fornece uma ferramenta eficaz para reduzir o uso desnecessário de memória e torna-o mais econômico. (Prakash et al., 2015)

Considerando todos os pontos anteriores, escolheram-se algumas tecnologias de *backend e frontend*, conforme são discriminadas na sequência do texto.

### 4.3.1 Seleção de tecnologias backend

Passa-se a apresentar um conjunto de tecnologias *backend* que serviram de análise comparativa para a escolha mais apropriada ao projeto.

#### Node.js<sup>93</sup>

O Node.js é um ambiente de tempo de execução JavaScript de código aberto, multiplataforma e executado no mecanismo V8<sup>94</sup> e executa código JavaScript fora de um navegador da Web, projetado para criar aplicações de rede escaláveis. O Node.js permite que os desenvolvedores usem JavaScript para escrever ferramentas de linha de comando e para scripts do lado do servidor para produzir conteúdo dinâmico da página da Web antes que a página seja enviada ao navegador da Web do usuário.

---

<sup>90</sup> É a estrutura que possibilita a operação do sistema. A função de um backend está relacionada com servidores, bancos de dados, segurança, estrutura, gerenciamento de conteúdo

<sup>91</sup> é um software de servidor web multiplataforma gratuito e de código aberto. O Apache é desenvolvido e mantido por uma comunidade aberta de desenvolvedores sob os auspícios da Apache Software Foundation. <https://www.apache.org>

<sup>92</sup> <https://nginx.org/>

<sup>93</sup>NodeJS: <https://nodejs.org/en>

<sup>94</sup> O V8 é um mecanismo JavaScript gratuito e de código aberto desenvolvido pelo Chromium Project para os navegadores Google Chrome e Chromium.

### Cron<sup>95</sup>

É um utilitário de linha de comando que atua como um “agendador” de tarefas em sistemas operacionais do tipo Unix. Os usuários que configuram e mantêm ambientes de software usam o *Cron* para agendar tarefas (comandos ou *shell scripts*<sup>96</sup>), para serem executados periodicamente em horários, datas ou intervalos fixos. Ele normalmente automatiza a manutenção ou administração do sistema, embora sua natureza de uso geral o torne útil para coisas como download de arquivos da Internet e download de e-mail em intervalos regulares.

### Geospatial Data Abstraction Library (GDAL)<sup>97</sup>

GDAL é uma biblioteca de processamento para formatos de dados geoespaciais matriciais e vetoriais que é lançada sob uma licença de código aberto. Ela também vem com uma variedade de utilitários de linha de comando úteis para leitura e processamento de dados.

### Generic Mapping Tools (GMT)<sup>98</sup>

GMT é uma coleção de ferramentas de linha de comando de código aberto para manipular conjuntos de dados geográficos e cartesianos (incluindo filtragem, ajuste de tendência, grade, projeção etc.) superfícies iluminadas artificialmente e vistas em perspectiva 3D. Ele suporta muitas projeções e transformações de mapas e inclui dados de suporte, como litorais, rios e limites políticos e, opcionalmente, polígonos de países.

### Strapi<sup>99</sup>

Strapi é um “*CMS headless*”<sup>100</sup> que utiliza a tecnologia Node.js de código aberto. Um sistema de gerenciamento de conteúdo headless, é um sistema de gerenciamento de conteúdo somente *backend* que atua principalmente como um repositório de conteúdo. Um CMS *headless* torna o conteúdo acessível por meio de uma API para exibição em qualquer dispositivo sem uma camada interna, *frontend* ou de apresentação.

---

<sup>95</sup> Cron. (2022, August 11). In Wikipedia. <https://en.wikipedia.org/wiki/Cron>

<sup>96</sup> É um programa de computador projetado para ser executado pelo sistema operacionais do tipo Unix, por um interpretador de linha de comando.

<sup>97</sup> GDAL/OGR Geospatial Data Abstraction software Library: URL <https://gdal.org>

<sup>98</sup> <https://www.generic-mapping-tools.org/about/>

<sup>99</sup> <https://strapi.io/>

<sup>100</sup> Um sistema de gerenciamento de conteúdo headless, ou CMS headless, é um sistema de gerenciamento de conteúdo da web somente *backend* que atua principalmente como um repositório de conteúdo

### Nginx<sup>101</sup>

Nginx é um servidor proxy reverso<sup>102</sup> e HTTP leve e de alto desempenho, bem como um servidor proxy de uso geral (TCP/UDP/IMAP/POP3/SMTP). O servidor proxy reverso<sup>102</sup>, que é caracterizado pelo baixo uso de memória e forte capacidade de simultaneidade.

### MapTiler<sup>103</sup>

MapTiler (maptiler.com) é um provedor global de serviços de mapas e uma empresa de desenvolvimento de software. A equipe MapTiler é o autor original e o principal colaborador do projeto OpenMapTiles, MapLibre e outros projetos de código aberto. A empresa serve gratuitamente mapas vetoriais na web para servirem a projetos científicos sem fins lucrativos.

#### 4.3.2 Seleção de Tecnologias Frontend

A aplicação web é desenvolvida com base nas principais tecnologias da Web – HTML para visualizar o conteúdo das páginas, CSS para estilo e formatação das páginas e Javascript suportados pela biblioteca React para habilitar a interatividade das páginas web. A lista de tecnologias é seguinte:

### React.js<sup>104</sup>

O React é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário (*frontend*) em páginas web. É mantido pelo Facebook, Instagram, outras empresas e uma comunidade de desenvolvedores individuais. O React pode ser usado como base no desenvolvimento de aplicações de página única (SPA – Single Page Application), móveis ou renderizados por servidor com outras componentes da biblioteca Node.js. O React atua sobre o gerenciamento de estado e renderização desse estado para o DOM<sup>105</sup> (*Document Object Model*).

---

<sup>101</sup> <https://nginx.org/>

<sup>102</sup> O servidor proxy reverso seleciona o servidor de destino para obter os dados e, em seguida, retorna ao cliente. O servidor proxy reverso e o servidor de destino são apenas um servidor para o mundo exterior, que expõe o endereço do servidor proxy e oculta o endereço IP do servidor real.

<sup>103</sup> <https://www.maptiler.com/>

<sup>104</sup> <https://react.dev/>

<sup>105</sup> é uma interface que representa como os documentos HTML e XML são lidos pelo seu browser. Após o browser ler seu documento HTML, ele cria um objeto que faz uma representação estruturada do seu documento e define meios de como essa estrutura pode ser manipulada.

### MapLibre<sup>106</sup>

MapLibre é um conjunto de SDKs (Kit de desenvolvimento de software) de renderização em Javascript usada para criar mapas interativos usando WebGL para desenvolvimento de aplicações móveis e da Web, existindo enquanto coleção de bibliotecas de mapeamento de código aberto gerenciada por uma comunidade de desenvolvedores.

### Bootstrap<sup>107</sup>

É um *framework* CSS de código aberto usado principalmente para facilitar e agilizar o design amigável para dispositivos móveis. O Bootstrap também atua como um provedor de ampla variedade de componentes, como botões, listas suspensas, formulários etc. tornando o projeto mais fácil de ser implementado em diversos dispositivos por terem recursos de RWD (*Responsive Web Design*).

### Material UI<sup>108</sup>

Material UI é uma biblioteca de componentes React de código aberto que implementa o Material Design<sup>109</sup> do Google. Ela será utilizada neste projeto para os componentes para dispositivos móveis e responsivos.

## 4.4 Projeto de Arquitetura

O projeto de arquitetura de sistemas é um processo criativo no qual você projeta uma organização de sistema que satisfará os requisitos funcionais e não funcionais de um sistema. Não existe um processo de projeto arquitetônico estereotipado padrão. Depende do tipo de sistema que está sendo desenvolvido, do histórico e experiência do arquiteto do sistema e dos requisitos específicos do sistema (Sommerville, 2016).

A arquitetura da aplicação Web pode se referir a vários componentes de design do sistema e suas interações. Uma arquitetura de alto nível com os principais componentes escolhidos para implementar os requisitos funcionais são definidos na seção 4.1 deste capítulo.

---

<sup>106</sup> <https://maplibre.org/>

<sup>107</sup> <https://getbootstrap.com/>

<sup>108</sup> <https://mui.com/material-ui/getting-started/>

<sup>109</sup> Material Design é uma linguagem de design livre desenvolvida pela Google com o objetivo de unificar e padronizar as interfaces gráficas dos sistemas da empresa,

Um diagrama de alto nível e um diagrama com os componentes foi projetada para que as partes envolvidas no projeto entendam melhor a integração dos diferentes softwares; Estes diagramas servem para: Comunicação e Compreensão; Tomada de Decisão; Documentação; Colaboração; Avaliação da Escalabilidade e Desempenho. As figuras seguintes ajudam a criar uma visão geral clara do sistema de software e orienta as decisões de design e comunica a estrutura do sistema para todas as partes interessadas e envolvidas no projeto.

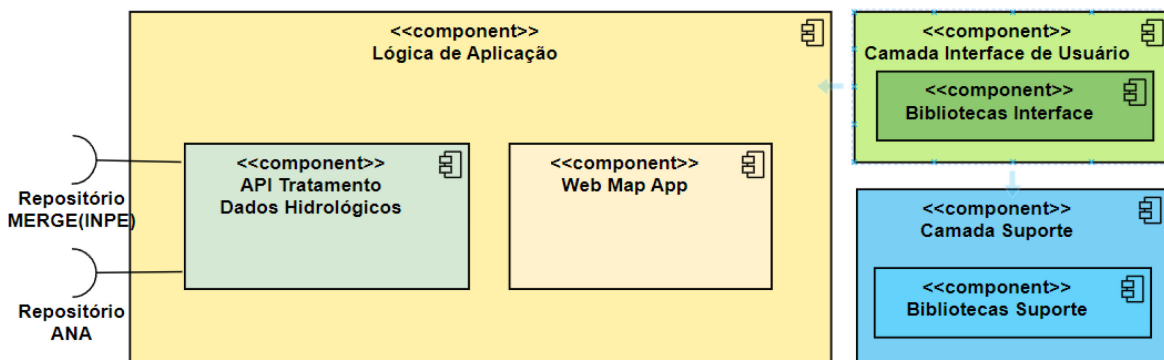


Figura 4.2: Arquitetura de alto nível do aplicativo.

As noções de separação e independência ajudam o projeto arquitetônico porque permitem que as mudanças sejam localizadas e organiza o sistema em camadas, com funcionalidades relacionadas associadas a cada camada. Uma camada fornece serviços para a camada acima dela (Sommerville, 2016).

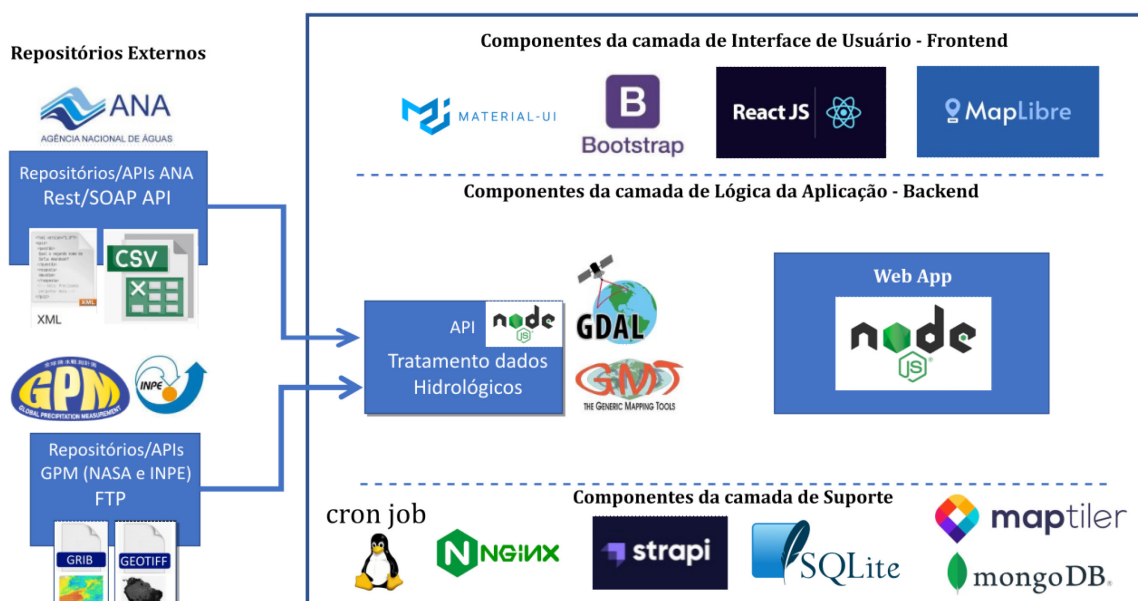


Figura 4.3: Diagrama de arquitetura com componentes especificados.

A camada mais baixa é a camada de suporte, representa os serviços básicos que serão utilizados pela camada acima. Inclui software de suporte ao sistema, suporte a banco de dados, servidor web, servidor de mosaico de mapas e sistema operacional com seus serviços. Na camada de suporte o componente Nginx atua como um ponto de entrada para requisições externas ao sistema, como um servidor proxy (representante<sup>110</sup>).

A próxima camada é a camada de aplicação ou lógica, que inclui os componentes relacionados com a funcionalidade da aplicação e os componentes utilitários usados por outros componentes do aplicativo.

Neste projeto, a camada de Lógica pode ser dividida em dois componentes principais, a primeira é responsável pela lógica para descarregar e processar os dados hidrológicos e a segunda pela geração das visualizações a partir dos dados armazenados na camada de banco de dados.

Quanto à camada de visualização no projeto, utiliza-se o *framework* React, importando diversas bibliotecas onde entre as principais: o MapLibre, responsável pelas funções de visualização dos mapas, e o *Bootstrap* que torna a aplicação Responsivo para diversos tamanhos de janela de visualização.

---

<sup>110</sup> Tradução nossa.

## Componente do Servidor (*Backend*) e Tratamento de Dados

Após a especificação dos requisitos e projeto de arquitetura, serão descritas, neste capítulo, a especificação e a implementação da camada que abastecerá, com os dados hidrológicos, a aplicação web, para que os requisitos funcionais sejam cumpridos.

Os dados tratados e processados pelo sistema provêm de dois repositórios distintos. O repositório da Agência Nacional de Águas e Saneamento (ANA – Brasil) contém os dados de precipitação diários coletados das estações pluviométricas do Brasil. O repositório é o Portal Hidroweb<sup>111</sup>, que é uma ferramenta integrante do Sistema Nacional de Informações sobre Recursos Hídricos<sup>112</sup> (SNIRH) e oferece o acesso ao banco de dados que contém todas as informações coletadas pela Rede Hidrometeorológica Nacional<sup>113</sup> (RHN), reunindo dados de níveis fluviais, vazões, chuvas, climatologia, qualidade da água e sedimentos.

O segundo repositório utilizado é o do Instituto Nacional de Pesquisas Espaciais<sup>114</sup> (INPE) do Brasil. O produto desse repositório é gerado e disponibilizado operacionalmente pelo CPTEC/INPE<sup>115</sup> (Centro de Previsão de Tempo e Estudos Climáticos) com a estimativa de precipitação por satélite utilizando os dados do *Global Precipitation Measurement*<sup>116</sup> (GPM) *Integrated Multi-satellite Retrievals for GPM*<sup>117</sup> (IMERG).

Na seção 5.1, descrevem-se a aquisição e do tratamento dos dados *in situ* dos repositórios da Agência Nacional de Águas e Saneamento. Na seção 5.2, discorre-se sobre a aquisição e o tratamento dos dados *in situ* dos repositórios do CPTEC/INPE.

Na seção 5.3, define-se uma API REST para armazenar e disponibilizar os dados hidrológicos para a visualização na camada de interface de usuário.

---

<sup>111</sup> <https://www.snirh.gov.br/hidroweb/>

<sup>112</sup> <https://www.gov.br/ana/pt-br/assuntos/gestao-das-aguas/politica-nacional-de-recursos-hidricos/sistema-de-informacoes-sobre-recursos-hidricos>

<sup>113</sup> <https://metadados.snirh.gov.br/geonetwork/srv/api/records/f85dbf06-a869-414c-afc5-bb01869e9156>

<sup>114</sup> <https://www.gov.br/inpe/pt-br>

<sup>115</sup> <https://www.cptec.inpe.br/>

<sup>116</sup> <https://gpm.nasa.gov/>

<sup>117</sup> <https://gpm.nasa.gov/data/imerg>

## 5.1 Aquisição e processamento dados hidrológicos in situ

A ANA disponibiliza, no seu repositório<sup>118</sup>, o acesso aos arquivos em formato CSV (valores separados por vírgula) de cada uma das mais de 16.000 estações cadastradas no seu sistema. Esses arquivos são processados por nossa API para tratamento de dados utilizando uma API desenvolvida em NodeJS<sup>119</sup> para analisar os ficheiros que foram descarregados em formato CSV, observando-se que cada linha corresponde a um mês de precipitação, conforme ilustra o exemplo disposto na Figura 5.1.

The screenshot shows the ANA Hidroweb v3.2.6 interface. The main content area is titled 'Séries Históricas de Estações' and includes a search form with fields for 'Tipo Estação', 'Cód', 'Bacia', 'Subt', 'Estado', and 'Mun'. Below the search form is a table with columns for 'Data', 'TipoMedica', 'Maxima', 'Total', 'DiaMaxima', 'NumD', 'Maxima', 'TotalStatus', 'NumDiasDeC', 'TotalAnual', 'TotalAr', 'Ch', 'Chuva03', 'Chuva04', 'Chuva05', 'Chuva06', 'Chuva07', and 'Ch'. The table contains data for various dates from 2020 to 2021.

13	Data	TipoMedica	Maxima	Total	DiaMaxima	NumD	Maxima	TotalStatus	NumDiasDeC	TotalAnual	TotalAr	Ch	Chuva03	Chuva04	Chuva05	Chuva06	Chuva07	Ch	
14	01/05/2021	1					0	0	0	0	0	0							
15	01/04/2021	1					0	0	0	0	0	0							
16	01/03/2021	1					0	0	0	0	0	0							
17	01/02/2021	1					0	0	0	0	0	0							
18	01/01/2021	1	129	9	881	8	21	23	1	1	1	0	54	1	0	0	0	38	
19	01/12/2020	1	51	4	442	8	23	21	1	1	1	0	0	0	0	0	0	39	
20	01/11/2020	1	50	6	376	4	2	16	1	1	1	0	39	3	50	6	10		
21	01/10/2020	1	18	6	106	4	6	10	1	1	1	0	10	0	7	6	0		
22	01/09/2020	1	38	9	86	9	16	8	1	1	1	0	0	0	0	0	0		
23	01/08/2020	1	34	2	127	7	28	13	1	1	1	0	0	0	0	0	3	0	
24	01/07/2020	1	59	1	246	5	1	28	1	1	1	0	59	1	8	2	10		

Figura 5.1: Repositório e extrato de arquivo CVS do repositório da ANA.

Dos vários métodos utilizados do *framework NodeJS* selecionamos para exemplo na Figura 5.2, o método *createReadStream*, que é uma interface de programação embutida no módulo FS do NodeJS, pode-se abrir uma pasta para ler os dados nela presentes, onde cada linha dos arquivos CSV (valores separados por vírgulas) é lida, e adicionada, ao objeto, uma estrutura de dados, sendo, então, inseridos na API REST da aplicação web.

<sup>118</sup> <https://www.snirh.gov.br/hidroweb/serieshistoricas>

<sup>119</sup> <https://nodejs.org/>

```

const stream = fs.createReadStream(filePath);
stream
  .pipe(parse({ delimiter: ';', from_line: 14 }))
  .on('data', function (csvrow) {
    monthRain = new MonthPrecipitation(csvrow[2], 1);
    ...
    for (var i = 13; i < daysInMonth + 13; i++) {
      monthRain.insertValue(csvrow[i]);
    }
    rainData._precipitation.push(monthRain);
  });
  ..
return new Promise((resolve, reject) => {
  stream.on('end', function () {
  ..

```

Figura 5.2: Extrato de código para processar dados de precipitação.

## 5.2 Aquisição e processamento dados de satélite

O INPE/CPTEC disponibiliza, em um de seus repositórios<sup>120</sup>, os arquivos de precipitação em formato GRIB, que é um formato de arquivo para o armazenamento e transporte de dados meteorológicos em grade. Os arquivos têm resolução de horizontal de 0,1° grau, com um corte de abrangência para América do Sul.

Os dados de sensoriamento remoto coletados pelo INPE/CPTEC são usados de duas formas mais operacionais neste projeto, a saber:

- Visualização de um mapa de calor de precipitação dos últimos sete dias da América do Sul - O mapa de calor de precipitação da América do Sul é gerado com o componente HeatMap.js, que, através de uma série de transformações na imagem GRIB gerada pelo INPE/CPTEC, utiliza bibliotecas dos Sistemas de Informação Geográfica GMT e GDAL. A imagem fonte da INPE/CPTEC do formato GRIB é convertida e recebe a aplicação uma palheta de cores para produzir o mapa de calor com os seguintes comandos GDAL/GMT: *gdal\_translate*<sup>121</sup>; *gmt grd2cpt*<sup>122</sup>; *gmt grdimage*<sup>123</sup>; *gmt psconvert*. Na Figura 5.3, tem-se o diagrama de sequência dos passos utilizados.

<sup>120</sup> <http://ftp.cptec.inpe.br/modelos/tempo/MERGE/GPM/DAILY/>

<sup>121</sup> [https://gdal.org/programs/gdal\\_translate.html](https://gdal.org/programs/gdal_translate.html)

<sup>122</sup> <http://gmt.soest.hawaii.edu/doc/latest/grd2cpt.html>

<sup>123</sup> <https://docs.generic-mapping-tools.org/latest/grdimage>

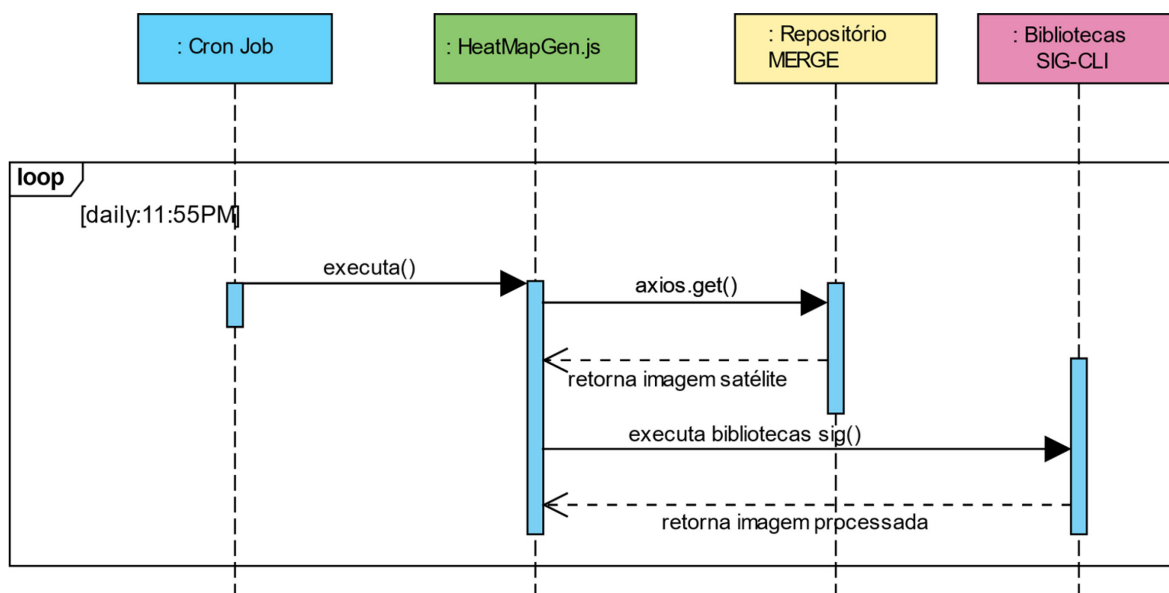


Figura 5.3: Diagrama de sequência para geração do mapa de calor de precipitação.

- Precipitação diária desde 2001, ano do começo da missão global *Precipitation Measurement (GPM) Integrated Multi-satellite Retrievals for GPM (IMERG)* para comparação com a aplicação web mediante o uso dos dados de precipitação coletados *in situ* da ANA. Conforme mostrado na Figura 5.4, o componente `gridData.js` é utilizado para gerar os dados de precipitação armazenados no banco de dados da API REST e visualização pela interface do nosso sistema.

```

readStations = () => {
  var stations = [];
  try { const data = fs.readFileSync(
    dirPathStations + `stations-${STATE}.json`,
    { encoding: "utf8", flag: "r" }
  )
  ...
  while (processtime < lastTime) {
    // create Date object based on the current process Time
    var pTime = new Date(processtime);
    var monthRainArray = [];
    for (var i = 0; i < stations.length; i++) {
      let mn = new MonthPrecipitation(processtime, 2);
      monthRainArray.push(mn);
    }
    ...
    for (var i = 1; i <= daysInMonth; i++) {
      ...
      var result = execSync(`gmt grdtrack ${DIR_COORD}stations-
        ${STATE}-coord.xyg -G${mergePath}${year}-${monthString}-
        ${dayString}.grib2 -Z`);
    }
  }
}

```

Figura 5.4: Extrato de código processar precipitação de satélite.

### 5.3 API REST – STRAPI

Para simplificação e gerenciamento do conteúdo de precipitação, foi implementado o CMS (*Content Management System*<sup>124</sup>) Strapi, que é Headless<sup>125</sup>. Essa configuração permite fornecer conteúdo para qualquer *frontend* ou interface de usuário. Ele permite a separação entre o gerenciamento e armazenamento de conteúdo e a apresentação desse mesmo conteúdo. A Figura 5.5 é uma composição de duas imagens. O lado esquerdo da Figura 5.5 apresenta as estações pluviométricas inseridas na API, enquanto, do lado direito, pode-se identificar um trecho dos campos das estações.

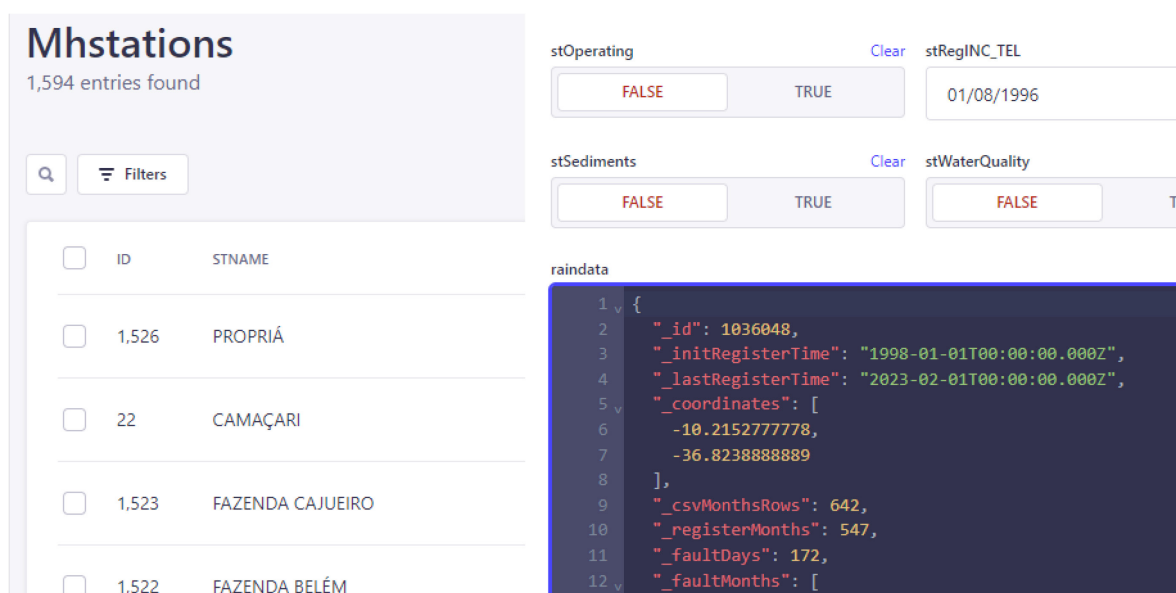


Figura 5.5: Estações pluviométricas do MapHidro no STRAPI.

Para facilitar as operações CRUD<sup>126</sup> e edição das estações, foi adicionado um subdomínio: <https://api.mahidro.com> para manipular o conteúdo com as rotas (*endpoints*)<sup>127</sup> ao realizar as requisições à API STRAPI. No contexto de REST (*Representational State*

<sup>124</sup> (CMS), que significa "Sistema de Gerenciamento de Conteúdo" em português, é uma plataforma de software que permite a criação, edição, publicação e gerenciamento de conteúdo digital, como textos, imagens, vídeos e outros tipos de informações, em um ambiente online.

<sup>125</sup> <https://strapi.io/> - Em um Headless CMS, o foco está exclusivamente no gerenciamento de conteúdo. Isso significa que o CMS se concentra em fornecer ferramentas para criar, editar e organizar conteúdo, mas não se preocupa com a forma como esse conteúdo é apresentado ao usuário final. A apresentação do conteúdo é delegada para uma camada separada, que pode ser uma aplicação web, um aplicativo móvel, um site estático ou qualquer outro tipo de plataforma digital.

<sup>126</sup> CRUD é um acrônimo que se refere a quatro operações básicas usadas em sistemas que interagem com bancos de dados. Essas operações são *Create* (Criação), *Read* (Leitura), *Update* (Atualização) e *Delete* (Exclusão).

<sup>127</sup> A estrutura de um *endpoint* geralmente segue o padrão da URL base do serviço, seguida por um caminho que identifica o recurso específico ou a ação a ser realizada.

*Transfer*) API, essas operações são mapeadas para as operações HTTP correspondentes. A Tabela 5.1 mostra as opções rotas padrões REST em relação as operações CRUD.

Tabela 5.1: Rotas (endpoints strapi)

Método	URL	Descrição
GET	/api/:pluralApiId	Listar todos os itens
POST	/api/:pluralApiId	Criar entrada
GET	/api/:pluralApiId/:documentId	Listar entrada
PUT	/api/:pluralApiId/:documentId	Atualizar entrada
DELETE	/api/:pluralApiId/:documentId	Apagar entrada

No excerto do código abaixo, tem-se a função de inserção da precipitação na API com as rotas, sendo as variáveis `BASE_URL` e `COLLECTION_NAME`. Utiliza-se um token de autenticação *Bearer*<sup>128</sup> para proteger os dados.

```
function processItem(code) {
  return new Promise((resolve) => {
    setTimeout(async () => {
      try {
        const headers = {
          Authorization: `Bearer ${TOKEN}`,
          'Content-Type': 'application/json',
        };
        const content = await readStreamAsync(code);
        const response = await axios.post(
          `${BASE_URL}/${COLLECTION_NAME}`,
          {
            data: content,
          },
          { headers }
        );
        resolve(response);
      } catch (e) {
        console.log(e);
      }
    }, 1000); // Delay in milliseconds
  });
}
```

Figura 5.6: Excerto de código para inserção de precipitação na API.

<sup>128</sup> Um token Bearer em JavaScript refere-se a um tipo de token de autenticação usado para proteger recursos de uma aplicação web ou API. Este token é chamado de "Bearer" porque a entidade que detém o token (geralmente, um usuário ou um aplicativo) apresenta-o ao servidor como prova de sua identidade para acessar recursos protegidos.

## Interface de Usuário e Desenvolvimento do Protótipo

Após a seção de desenvolvimento do lado do servidor (*backend*) e tratamento de dados, na seção 6.1, são descritos o desenvolvimento do protótipo da interface de usuário (*frontend*) com técnicas de design de experiência de usuário (UX) para criar e refinar a interface de usuário do protótipo. Na seção 6.2, descrevem-se os detalhes da implementação física do protótipo. Na seção 6.3, algumas telas reais da aplicação web são visualizadas utilizando design responsivo para visualização em dispositivos móveis.

## 6.1 UX (Experiência de Usuário) para interface de usuário

As técnicas de UX foram utilizadas para refinar a interface de usuário. Para entender as necessidades dos usuários, foram conduzidas entrevistas abertas com alguns perfis de profissionais que utilizam dados hidrológicos/meteorológicos para entender e projetar para as necessidades e objetivos do público-alvo.

As técnicas de design de experiência de usuário utilizadas dependem do contexto do projeto e das necessidades específicas dos potenciais usuários. “O processo de design de UX é, geralmente, iterativo, com revisões e refinamentos contínuos com base no *feedback* dos usuários e na evolução das necessidades” (Benyan, 2019).

Tabela 6.1: Técnicas de experiência de usuário

1	<b>Pesquisa de Usuário:</b> Entrevistas com usuários foram realizadas para compreender suas necessidades, comportamentos e expectativas.
2	<b>Personas:</b> Foram criadas personas fictícias que representam perfis de usuários com base nos dados coletados na pesquisa.
3	<b>Jornadas do Usuário:</b> As jornadas são mapeadas através de fluxos de interação do início da tarefa até a conclusão da tarefa ou objetivo.
4	<b>Wireframes:</b> Representações visuais de baixa fidelidade para esboçar uma estrutura e layout do sistema para planejar a disposição dos elementos da interface.
5	<b>Design Responsivo:</b> Garantir que a interface seja compatível com diferentes dispositivos e tamanhos de tela, proporcionando uma experiência consistente.
6	<b>Feedback contínuo:</b> Durante o desenvolvimento foi aberto um canal contínuo através de novas conversas com os potenciais usuários e orientadores da dissertação visando realizar melhorias contínuas.

Fonte: adaptado de Benyan (2019).

Após as entrevistas com os usuários potenciais, foram desenvolvidos os seguintes passos: *personas*, *jornada de usuário* (fluxo de usuário) e *wireframes*.

Foram criadas duas personas mediante a análise das entrevistas<sup>129</sup> com as partes interessadas. “As personas são uma ferramenta valiosa para humanizar a pesquisa de usuário, orientar o design e garantir que os produtos e serviços atendam às necessidades e expectativas dos usuários reais. Elas são um componente essencial no processo de design de UX e podem ser usadas ao longo de todo o ciclo de desenvolvimento de produtos.” (Benyan, 2019).



Figura 6.1: Personas criadas com entrevistas

Na Tabela 6.2, tem-se um cenário de usuário. Um cenário de usuário é uma narrativa que descreve uma situação fictícia em que um usuário interage com o produto ou serviço. Essa configuração ajuda a visualizar como os usuários podem interagir com o sistema em um contexto do mundo real.

<sup>129</sup> Alguns trechos das entrevistas abertas podem ser vistos no Capítulo 4.

Tabela 6.2: Cenário de um usuário.

OBJETIVO	COLETA DE INFORMAÇÃO	ANÁLISE DE TAREFA
Um engenheiro necessita analisar dados de precipitação de uma área para planejar dimensionamento de obras de drenagem.	<ul style="list-style-type: none"> <li>• Comparar os dados de satélite com dados <i>in situ</i>.</li> <li>• Verificar a consistência dos dados rapidamente, para posterior processamento com ferramentas científicas.</li> <li>• Descarregar os dados de satélite e <i>in situ</i> para análise posterior.</li> <li>• Ele acessa esta informação em diferentes dispositivos: Notebook e tablets e smartphones.</li> </ul>	<p>Ponto de entrada: Acessar o App</p> <p>Critério de sucesso: visualizar dados de precipitação e comparar <i>in situ</i> x satélite</p> <hr/> <p>Tarefas podem ser visualizadas na Figura 6.2 e na Figura 6.3</p>

A Jornada de Usuário (Fluxos de telas da aplicação web) e Mapa de Navegação, têm o intuito de alinhar os caminhos e ações que o usuário pode fazer, dando sequência para o design com *wireframes* e para a implementação do design final da interface do

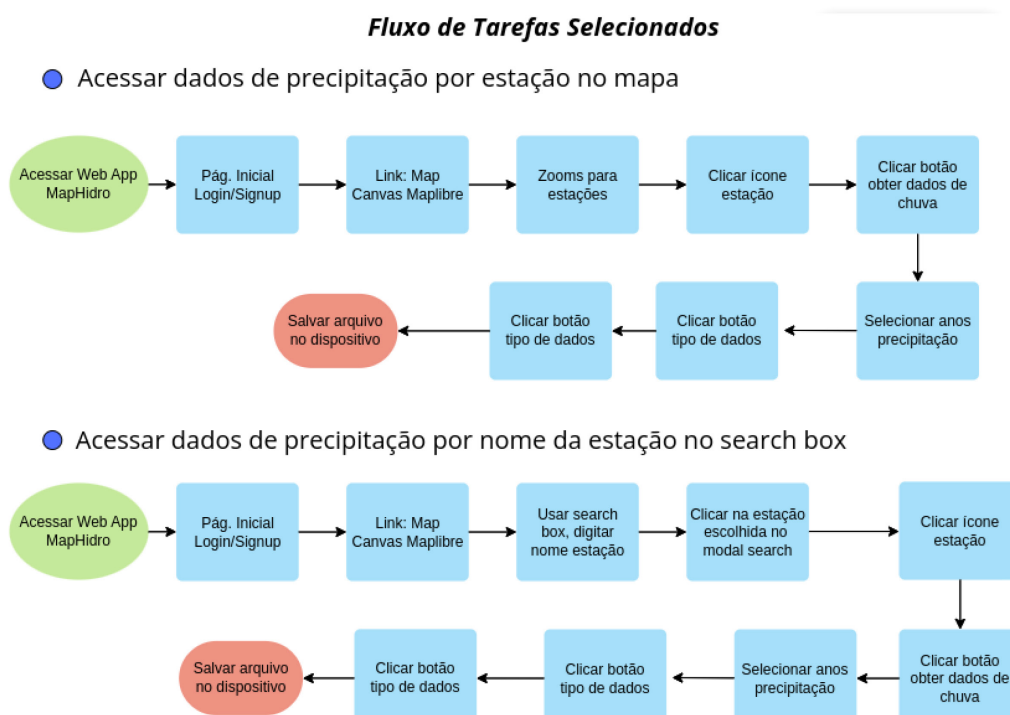


Figura 6.2: Fluxo de tarefas do sistema.

Foram desenvolvidos mapas de navegação, que segundo Benyon (2019) os mapas de navegação podem ser redesenhados de forma útil várias vezes ao longo do ciclo de vida do projeto, já que uma má estrutura de navegação é um dos principais motivos pelos quais as pessoas podem não utilizar um site, por exemplo. Os mapas podem ser usados com cenários para “percorrer” atividades específicas e são uma ótima maneira de detectar aspectos deficientes do design, como “páginas órfãs” (páginas que não são acessíveis), becos sem saída ou caminhos que são muito longos.

Seguindo a criação de mapas de navegação, iniciamos o design de *wireframes*. O *wireframes* fazem sentido ao serem usados para envolver membros da equipe em discussões e decisões de design, promovendo uma abordagem colaborativa e alinhada, para esboçar a estrutura e o layout da interface e desempenhar um papel importante na criação de designs eficazes.

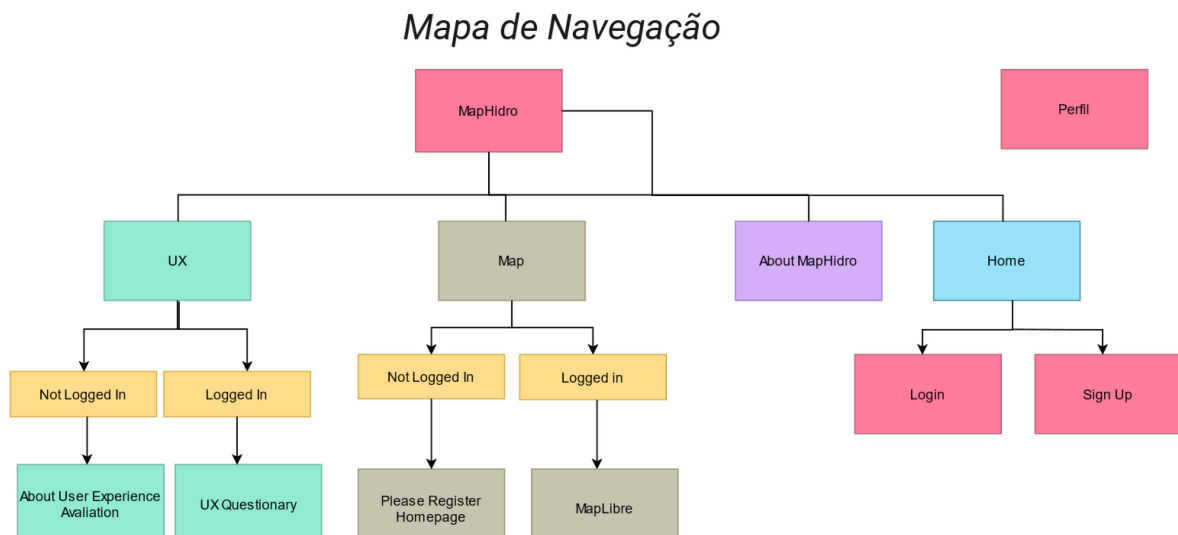


Figura 6.3: Mapa de navegação.

Para Benyon (2019), assim como os mapas de navegação concentram-se em como as páginas são estruturadas e vinculadas entre si, os *wireframes* concentram-se na estrutura de tipos específicos de páginas e na navegação entre as páginas. Usando os dois juntos formará o básico do design de uma aplicação web ou sítio web. Os wireframes funcionam porque concentram-se nos elementos gerais de um design sem a preocupação com os detalhes finais. Nas figuras seguintes, incluímos os designs de *wireframes* para o sistema quer para a versão desktop quer para a versão para dispositivos móveis.

Os *wireframes* abaixo ajudam na criação das interfaces dos seguintes casos de utilização: visualização de precipitação in situ mensal e precipitação por satélite mensal.

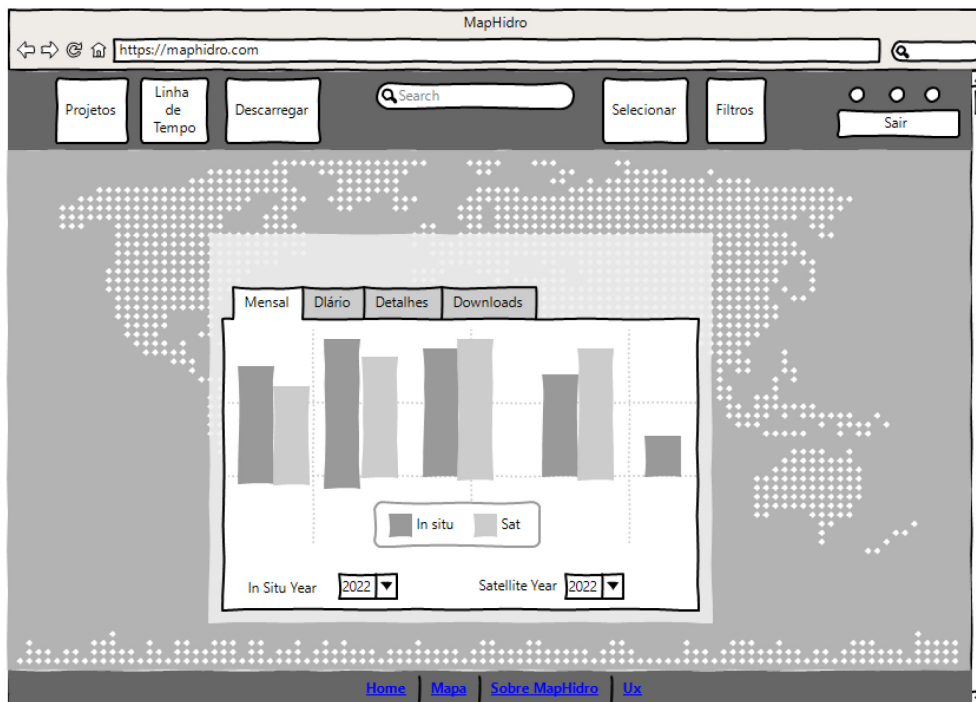


Figura 6.4: *Wireframe* Gráficos de Média de Chuva.

A figura a seguir ajuda criação da interface que implementará o caso de uso de busca por nome de estação.

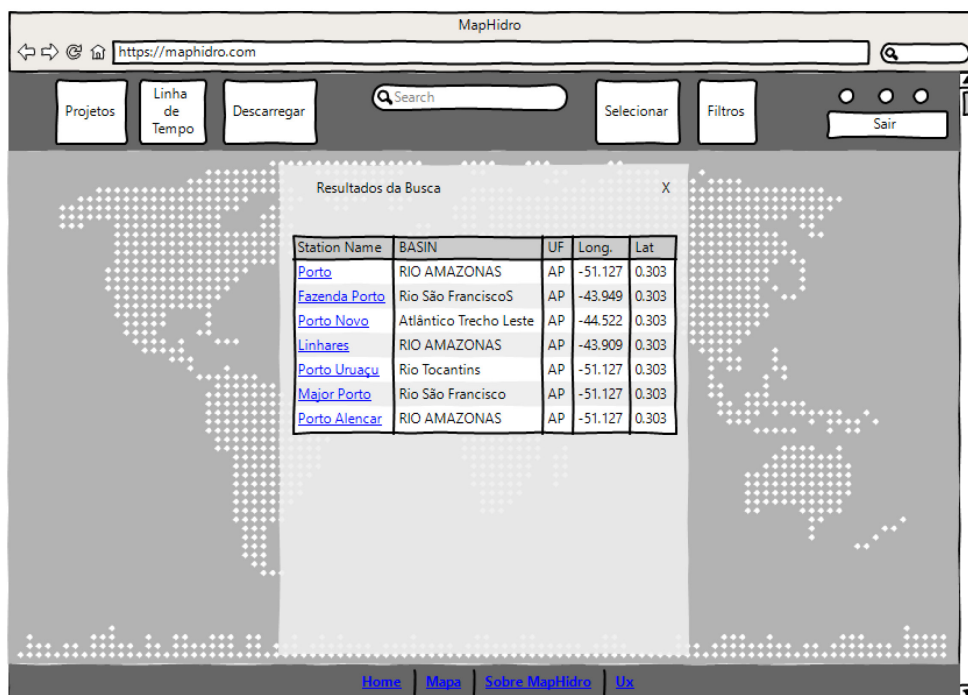


Figura 6.5: *Wireframe* Resultados de Busca.

Os *wireframes* a seguir nas figuras abaixo ajudam a desenvolver a interface para visualização compacta para dispositivos móveis. As interfaces para interatividade com os mapas em mosaicos são criadas, assim como também a caixa de diálogo de informações iniciais das estações e visualização dos dados processados.



Figura 6.6: *Wireframes* para dispositivo móvel

## 6.2 Implementação Física (Deployment)

A implementação foi realizada com a integração de diversas tecnologias de frontend como HTML, CSS, Javascript, React, entre outras bibliotecas de código livre, que foram

integradas com tecnologias de *backend* nos servidores web descritos nos capítulos anteriores. A interface visual do protótipo e suas funcionalidades são apresentadas nas figuras a seguir de acordo com os casos de uso e requisitos funcionais descritos no item 4.1 do capítulo quarto.

O protótipo da aplicação web necessita dos seguintes softwares da camada de suporte: Servidor HTTP Nginx, NodeJS, serviço Linux de tarefas agendadas (Cron), API Strapi, que armazena e retorna requisições dos dados hidrológicos. O serviço HTTP Nginx atua como um intermediário entre os clientes na Internet e os servidores web localizados na rede interna de uma organização. A sua principal função será receber as solicitações dos clientes e encaminhá-las para os servidores internos apropriados, retornando a resposta aos clientes. A imagem abaixo especifica a arquitetura física e lógica da camada de suporte:

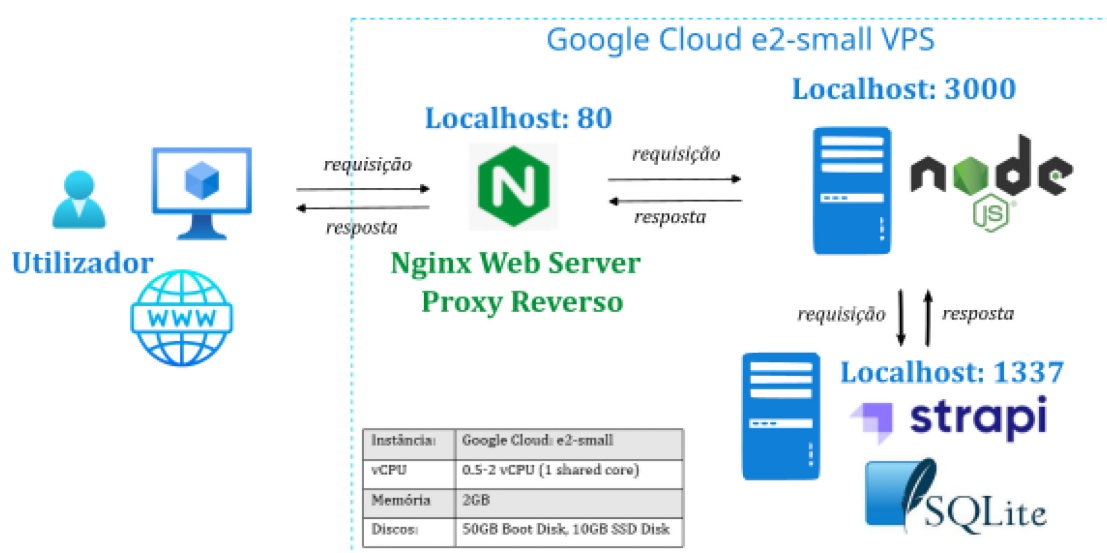


Figura 6.7: Arquitetura de implantação da camada de suporte

### 6.2.1 Implementação em ambiente de produção

O ambiente de produção em software é um ambiente de computação no qual um sistema de software ou aplicação é executado e disponibilizado para uso pelos usuários finais ou clientes. Esse ambiente é o estágio final no ciclo de desenvolvimento de software, onde o software é implantado e operacionalizado para atender às necessidades do público-alvo.

As boas práticas na implementação conduziram à aplicação do seguinte método: o código fonte do sistema é gerido pelo ambiente de desenvolvimento integrado (IDE) *Visual Studio Code*, que também está inserindo os arquivos na plataforma *Github*<sup>130</sup>. O GitHub é uma plataforma de hospedagem de código-fonte e colaboração para desenvolvimento de

<sup>130</sup> <https://github.com/>

software. Ela fornece um ambiente baseado na web que permite que desenvolvedores e equipes de desenvolvimento trabalhem juntos para criar, armazenar, gerenciar e compartilhar projetos de software.

Através da ferramenta *GitHub Workflows*, também conhecida como *GitHub Actions*, que é uma funcionalidade do *GitHub* que permite automatizar tarefas e fluxos de trabalho relacionados ao desenvolvimento de software, configuramos a automação de Compilação e Implantação: Configuramos um fluxo de trabalho para compilar o código e implantá-lo no ambiente de produção do servidor sempre que houver mudanças no repositório. Na figura 6.8 temos a página do projeto no *GitHub*.

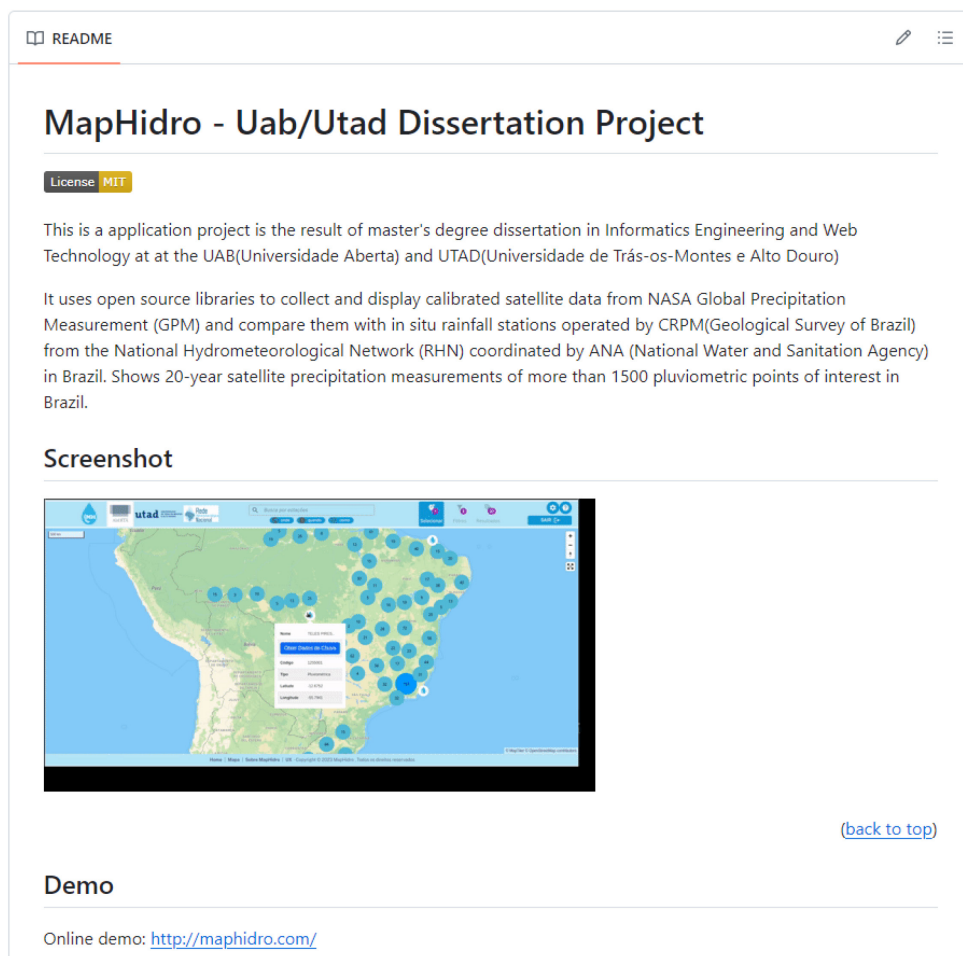


Figura 6.8: Github do projeto: <https://github.com/bmmoreira/maphidro-app>

### 6.3 Implementação de Design Responsivo

A interface da aplicação web utiliza em seus componentes o sistema de grades que se adaptam às diferentes resoluções de tela e tamanhos de dispositivos para proporcionar uma experiência de usuário consistente. Esse sistema de grades divide a largura da página em 12 colunas do *Bootstrap*<sup>131</sup> e *Material UI Breakpoints*<sup>132</sup>, permitindo que os elementos do layout sejam organizados em várias combinações, de forma que eles se ajustem automaticamente ao tamanho da tela.

Os componentes do MUI (Material UI) utilizam o sistema do Material Design<sup>133</sup> que é uma linguagem de design livre desenvolvida pela Google como o objetivo de unificar e padronizar as interfaces gráficas dos sistemas da empresa.

Em nosso protótipo, através de pontos de quebra(*breakpoints*) e detecção de largura de páginas apresentamos componentes diferentes de acordo com a largura. O excerto de código abaixo exemplifica a utilização de CSS (*Cascading Style Sheet*) de acordo com a largura configurada.

```
return (
  <>
    <div
      ref={rootRef}
      className={width < appSettings.mobileBreakpoint ?
        'map-wrap-mobile' : 'map-wrap'}>
      <div ref={mapContainerRef} className="map">
        {appState.modals.panelBox && (
          <PanelModals flyTo={flyToStation}
            onLayersHandleChange={layersHandleChange} />
        )}
      </div>
    </div>
  </>
)
} export default Map;
```

Figura 6.9: Excerto código componente Map.tsx

Adicionalmente outros componentes no sistema utilizam um recipiente(*container*) com pontos de quebra para ajustar-se a tela dos usuários, de acordo com a Tabela 6.3.

Tabela 6.3: Pontos de quebra para dimensionamento de componentes.

xs, extra-small: 0px	sm, small: 600px	md, medium: 900px	lg, large: 1200px	xl, extra-large: 1536px
-------------------------	------------------	----------------------	-------------------	----------------------------

<sup>131</sup> <https://getbootstrap.com/docs/4.0/layout/grid/>

<sup>132</sup> <https://mui.com/material-ui/customization/breakpoints/>

<sup>133</sup> <https://m3.material.io/>

As Figuras 6.10 a 6.15 são composições da visualização da aplicação web, onde, no lado esquerdo, apresenta-se a visualização para Desktop, enquanto, do lado direito, para dispositivo móvel.

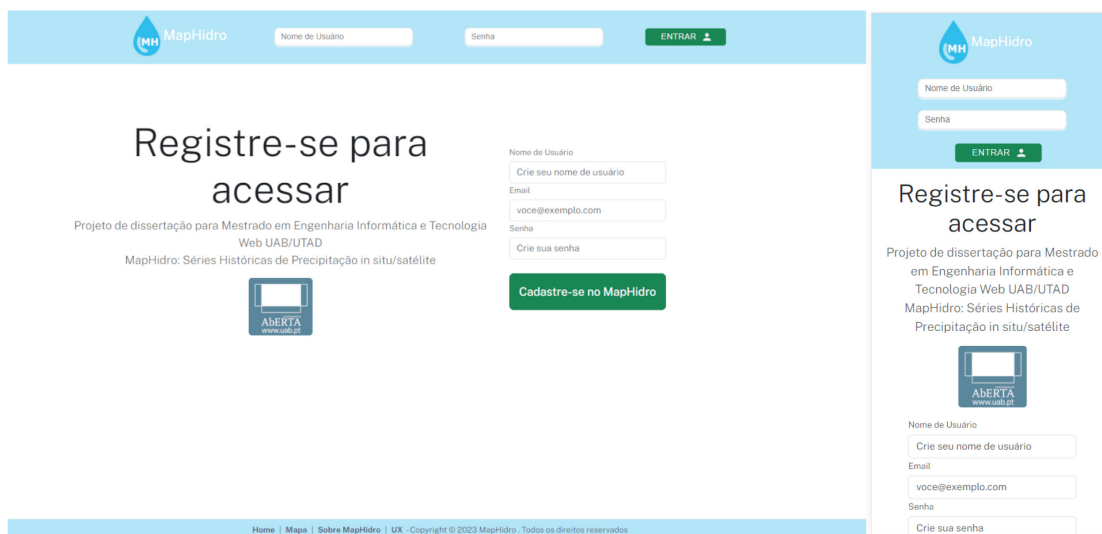


Figura 6.10: Protótipo da página inicial.

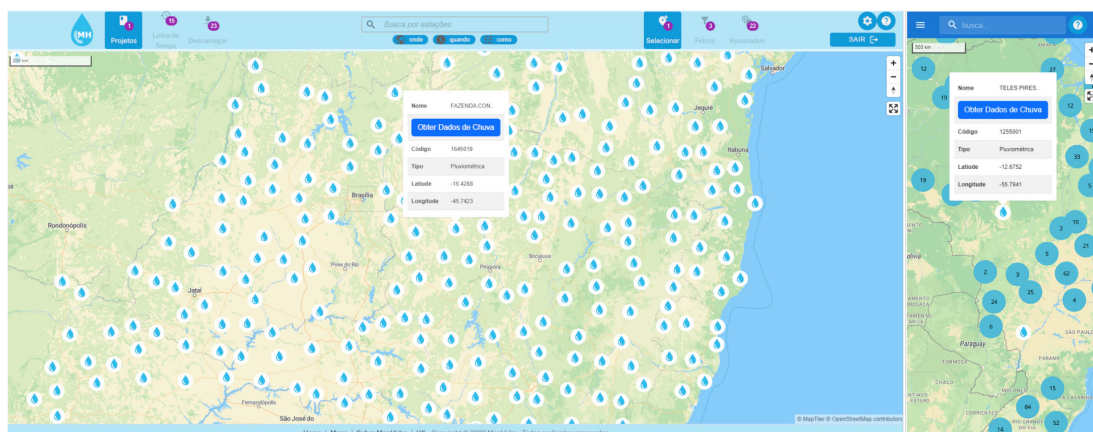


Figura 6.11: Janela *Pop-up* de estação.

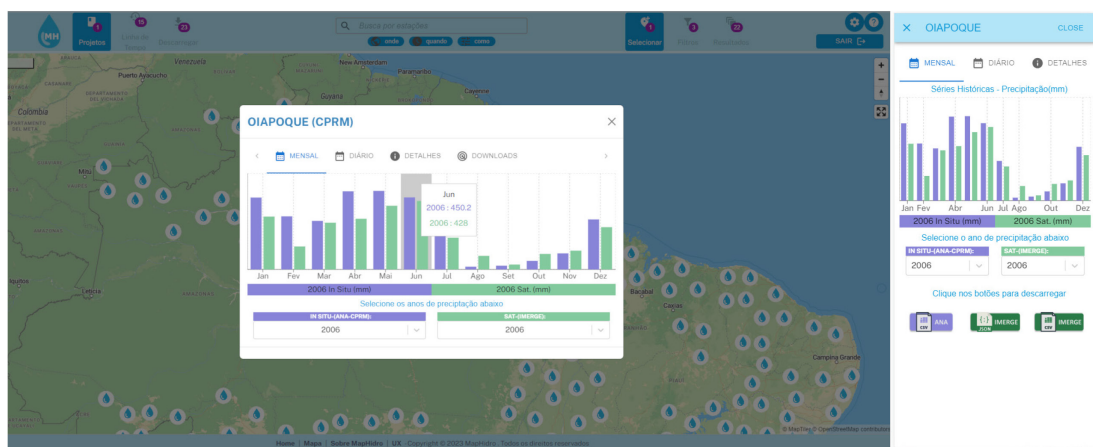


Figura 6.102: Precipitação mensal.

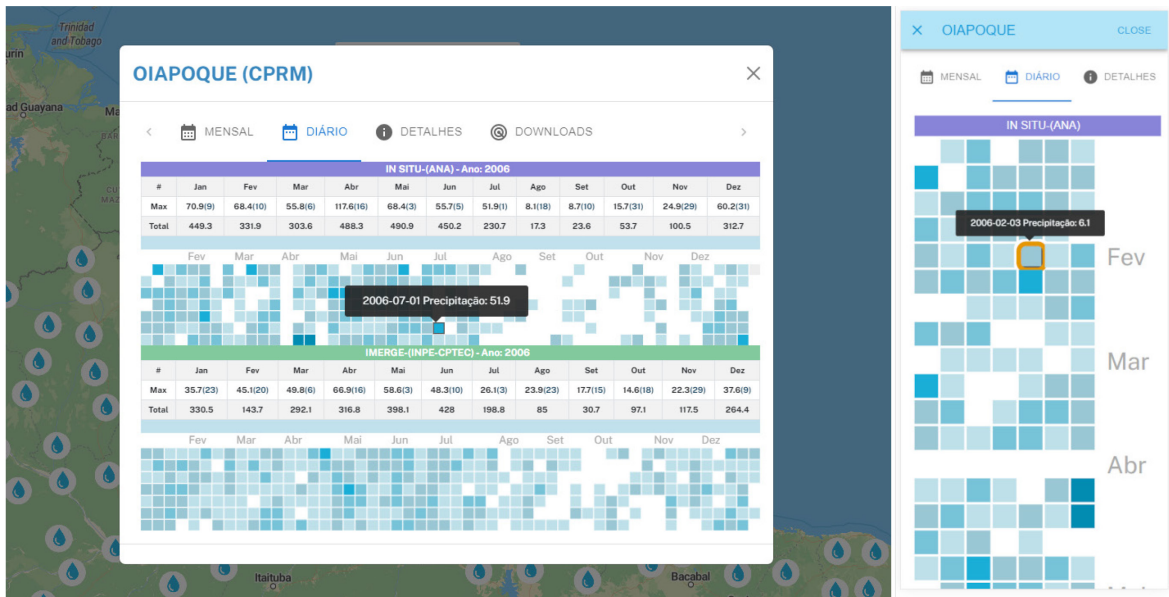


Figura 6.13: Precipitação diária.

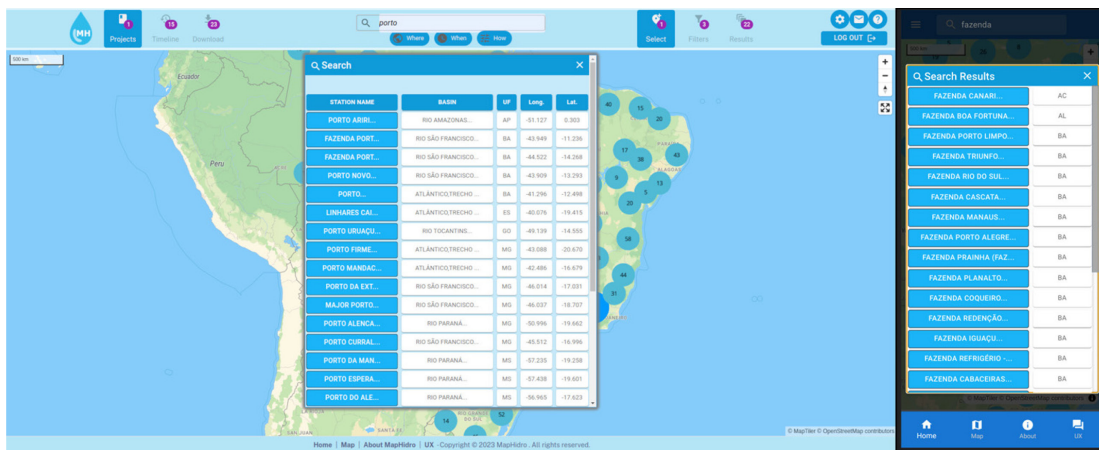


Figura 6.14: Resultado de busca por nome.

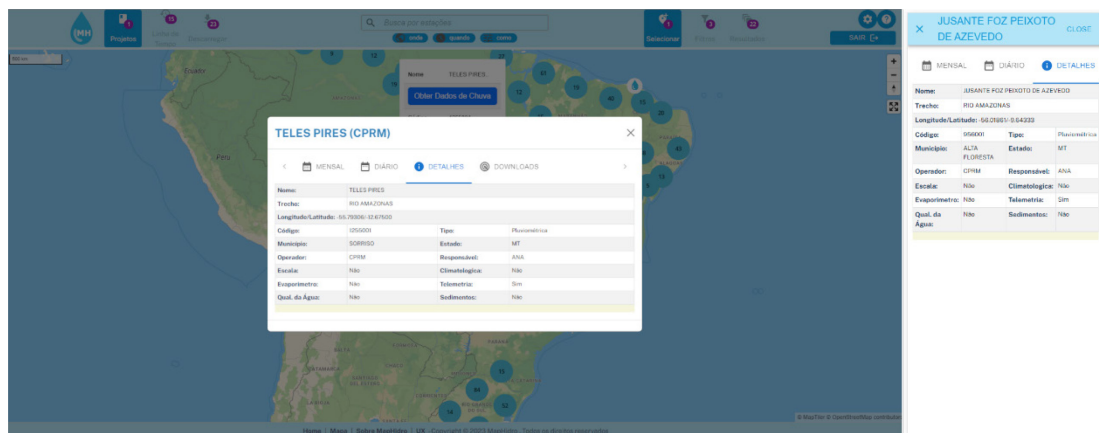


Figura 6.15: Aba detalhes de estação.

## Testes e Resultados

Neste capítulo, descrevem-se os testes realizados e a avaliação dos resultados obtidos com a pesquisa com usuários para verificar a satisfação e qualidade da aplicação web.

Na seção 7.1, apresentam-se os testes realizados. Os testes utilizados foram uma mescla de testes manuais e testes automatizados unitários. Os testes manuais são frequentemente usados em combinação com testes automatizados para obter uma cobertura abrangente de testes. Eles desempenham um papel na garantia da qualidade do software, especialmente quando se trata de aspectos subjetivos, como a usabilidade e a satisfação do usuário. Os testes unitários desempenham um papel crucial na garantia da qualidade do software, na facilitação da manutenção e na promoção de práticas de desenvolvimento eficazes. Eles são uma parte essencial da cultura de desenvolvimento de software voltada para qualidade e confiabilidade.

Na seção 7.2, dispõem-se os resultados do questionário de experiência de usuário utilizando a metodologia UEQ<sup>134</sup>. O UEQ é um questionário desenvolvido por um grupo de pesquisadores europeus (Laugwitz, B, et al.,2008), que conta com 26 termos relacionados à experiência do usuário e seus antônimos. O objetivo principal é capturar a experiência do usuário em termos de usabilidade, apelo estético e fatores emocionais ao interagir com um sistema, website, aplicação web ou produto. Ele fornece uma maneira de avaliar a satisfação e a qualidade da experiência do usuário. O questionário é composto por uma série de itens e escalas que os usuários preenchem após interagir com o sistema ou produto. Os resultados do UEQ podem fornecer informações valiosas para designers e pesquisadores, ajudando a melhorar a usabilidade e a experiência do usuário.

---

<sup>134</sup> <https://www.ueq-online.org/> -

## 7.1 Testes Manuais e Automatizados

No protótipo, utilizam-se testes manuais e automatizados. A Figura 7.1 descreve alguns pontos sobre os testes manuais e automatizados.



Figura 7.1: Diferenças de testes manuais e automatizados.

**Testes manuais:** foram aplicados testes manuais a cada nova iteração de desenvolvimento do sistema garantindo que os requisitos do software estavam sendo cumpridos.

**Testes automatizados:** teste unitários em componentes chaves do sistema com o framework Vitest<sup>135</sup>; foram criados testes unitários para a API de Login e Cadastro configurados como um subdomínio da aplicação web; a API usa a autenticação baseada em Token, onde guardamos o JSON Web Token (JWT) no lado do cliente; a sequência de Login pode ser visualizada na Figura 7.2.

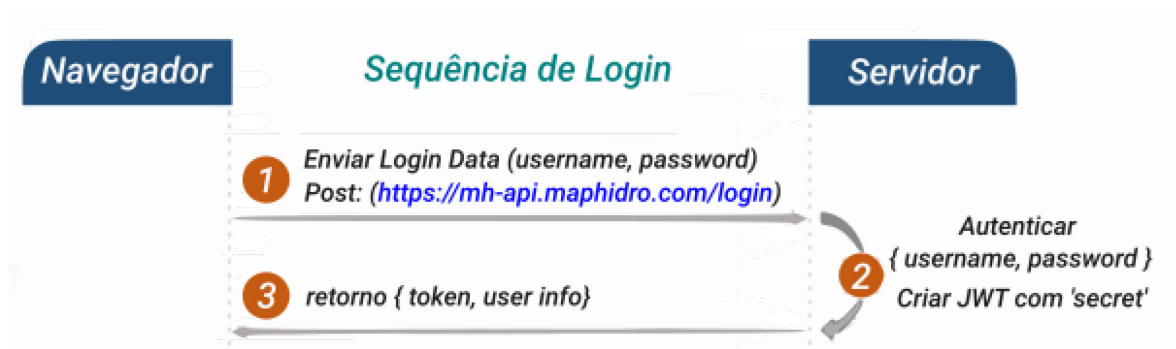


Figura 7.2: Esquema de sequência de login/registro no aplicativo.

<sup>135</sup> <https://vitest.dev/>

A função de Login procura pelo nome de usuário no banco de dados e compara a senha enviada pelo usuário com o *hash*<sup>136</sup> da senha no banco de dados, com o componente Bcrypt<sup>137</sup> do NodeJs. O excerto de código de teste para a função de login e os resultados dos testes podem ser visualizados na Figura 7.3.

```

1  /** @format */
2  import axios from 'axios';
3  import { it, expect, describe } from 'vitest';
4
5  describe('Login API', function () {
6      it('should return success on valid credentials', () =>
7          new Promise((done) => {
8              const credentials = {
9                  username: 'user',
10                 password: 'maphidroPASSWORD',
11             };
12
13             expect(makePostRequest(credentials)).resolves.toBeTruthy
14             done();
15         }));

```

```

> maphidro-backend@0.0.23 test
> vitest --run --reporter verbose

RUN v0.34.3 /home/bmoreira/projects/backend-api

✓ db.test.js (3) 1709ms
✓ Login API (3) 1709ms
  ✓ should return success on valid credentials 702ms
  ✓ should false on invalid credentials 450ms
  ✓ should return success the name of the user succesfull response 553ms

Test Files 1 passed (1)
Tests 3 passed (3)
Start at 19:55:20
Duration 2.18s (transform 30ms, setup 0ms, collect 55ms, tests 1.71s, en

```

Figura 7.3: Composição de excerto de código e resultado dos testes com Vitest.

<sup>136</sup> Função *hash*, é uma transformação matemática que converte um conjunto de dados em uma sequência alfanumérica de tamanho fixo. As senhas não são armazenadas em seu formato original nos sistemas de autenticação, para evitar exposição em caso de violações de segurança. Em vez disso, uma *hash* da senha é armazenada.

<sup>137</sup> O "Bcrypt" é uma adaptação da função de hash de senha "bcrypt" baseada no algoritmo Blowfish, que é projetada para ser lenta e deliberadamente intensiva em CPU, tornando muito difícil para atacantes descobrirem senhas usando ataques de força bruta.

## 7.2 Avaliação de experiência de usuário

Para avaliar a experiência de Usuário ao navegar pela aplicação web MapHidro foi utilizada a metodologia UEQ (Laugwitz, B, et al.,2008). Esta técnica permite uma avaliação rápida de qualquer produto interativo. As escalas do questionário são projetadas para cobrir uma impressão abrangente de experiência de usuário. O formato do questionário permite ao usuário expressar imediatamente sentimentos, impressões e atitudes que surgem quando eles usam um produto.

No total, vinte e quatro pessoas responderam ao questionário inserido no sistema.

As informações demográficas dos participantes na avaliação do software podem ser visualizadas nas Figuras 7.4, 7.5 e 7.6.

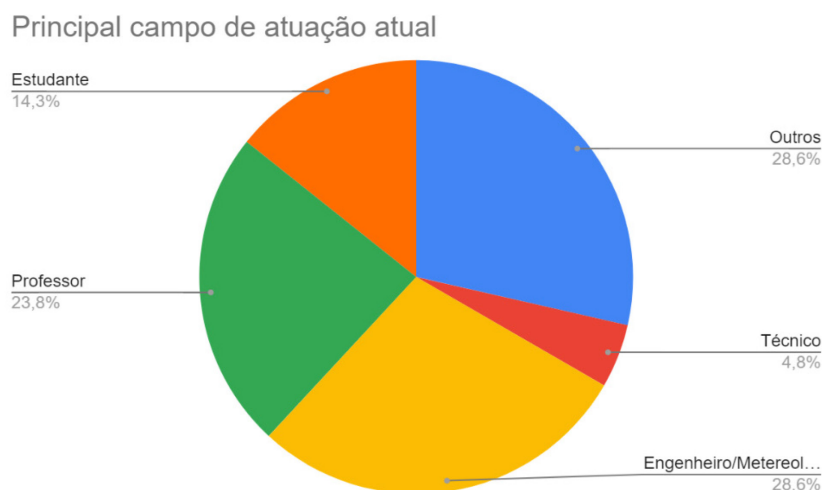


Figura 7.4: Principal campo de atuação atual.

Os perfis quanto à atuação profissional atual são diversos com percentagens aproximadas em dois deles, Professor e Engenheiro/Meteorologista, mas a amostra foi abrangente, pois foi seguida de respostas correspondentes ao conjunto de perfis de entrevistados na fase da análise de requisitos.

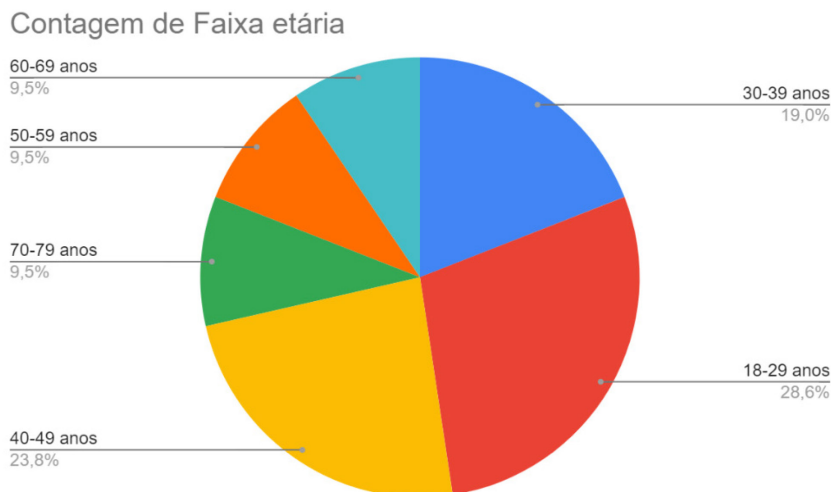


Figura 7.5: Faixa etária.

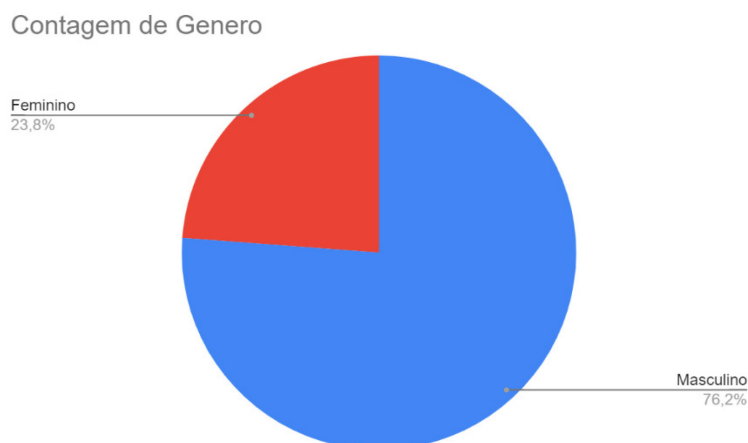


Figura 7.6: Faixa etária.

O questionário de experiência de usuário contém seis (6) escalas com 26 itens no total, e está disponível para visualização no apêndice I desta dissertação. As escalas estão alinhadas na Tabela 7.1.

Tabela 7.1: Escalas Metodologia UEQ

<b>1</b>	<b>Atratividade:</b> Impressão geral em relação ao produto. Os usuários gostam ou desgostam do produto. Itens: Agradável/Desagradável; Bom/Mau; Desinteressante/Atrativo; Incômodo/Cômodo; Atraente/Feio; Simpático/Antipático;
<b>2</b>	<b>Transparência:</b> É fácil entender como usar o produto? É fácil se familiarizar com o produto? Itens: Incompreensível/Compreensível; De Fácil aprendizagem/De difícil aprendizagem; Complicado/Fácil; Evidente/Confuso;
<b>3</b>	<b>Eficiência:</b> Ele é um produto rápido e eficiente? A interface é organizada? Itens: Rápido/Lento; Ineficiente/Eficiente; Impraticável/Prático; Organizado/Desorganizado;

4	<b>Controle:</b> O usuário sente-se em controle da interação? É a interação previsível? Itens: imprevisível/previsível; obstrutivo/condutor; seguro/ inseguro;
5	<b>Estimulação:</b> É interessante e excitante usar o produto? O usuário sente-se motivado para avançar no uso do produto? Itens: valioso/sem valor; aborrecido/excitante; desinteressante/interessante; motivante/desmotivante;
6	<b>Inovação:</b> É o design do produto inovativo e criativo? O produto chama a atenção do usuário? Itens: criativo/sem criatividade; original/convencional; comum/Vanguardista; conservador/Inovador

Fonte: Laugwitz et al. (,2008).

A análise das respostas dos usuários foi realizada com uma ferramenta fornecida pela metodologia UEQ<sup>138</sup>. A escala vai de entre -3 (muito mau) até +3 (muito bom); valores entre -0,8 e 0,8 representam uma avaliação mais ou menos neutra, enquanto valores maiores que 0,8 representam uma avaliação positiva, e valores menores -0,8, uma avaliação negativa.

Os resultados gerais de cada um dos itens do questionário podem ser visualizados na Tabela 7.2.

Tabela 7.2: Valores por itens individuais.

Item	Mean	Variance	Std. Dev.	No.	Left	Right	Scale
1	↑ 1.8	4.0	2.0	20	Desagradável	Agradável	Atractividade
2	↑ 1.8	3.3	1.8	20	Incompreensível	Compreensível	Transparência
3	↑ 1.7	3.6	1.9	20	Criativo	Sem criatividade	Inovação
4	↑ 3.0	0.0	0.0	20	De Fácil aprendizagem	De difícil aprendizagem	Transparência
5	↑ 1.7	2.9	1.7	20	Valioso	Sem valor	Estimulação
6	⇒ 0.6	2.4	1.5	20	Aborrecido	Excitante	Estimulação
7	↑ 1.5	3.5	1.9	20	Desinteressante	Interessante	Estimulação
8	↑ 1.0	3.2	1.8	20	Imprevisível	Previsível	Controlo
9	↑ 2.3	2.1	1.5	20	Rápido	Lento	Eficiência
10	↑ 1.3	4.5	2.1	20	Original	Convencional	Inovação
11	↑ 0.8	3.4	1.9	20	Obstrutivo	Condutor	Controlo
12	↑ 1.3	4.4	2.1	20	Bom	Mau	Atractividade
13	↑ 1.3	5.7	2.4	20	Complicado	Fácil	Transparência
14	↑ 1.9	1.8	1.3	20	Desinteressante	Atrativo	Atractividade
15	↑ 1.4	3.2	1.8	20	Comum	Vanguardista	Inovação
16	↑ 1.3	2.6	1.6	20	Incómodo	Cómodo	Atractividade
17	↑ 2.2	3.0	1.7	20	Seguro	Inseguro	Controlo
18	⇒ 0.6	2.7	1.6	20	Motivante	Desmotivante	Estimulação
19	↑ 1.8	2.6	1.6	20	Atende as expectativas	Não atende as expectativas	Controlo
20	↑ 1.7	2.4	1.6	20	Ineficiente	Eficiente	Eficiência
21	↑ 1.4	1.8	1.4	20	Evidente	Confuso	Transparência
22	↑ 1.1	3.1	1.8	20	Impraticável	Prático	Eficiência
23	↑ 1.9	2.0	1.4	20	Organizado	Desorganizado	Eficiência
24	↑ 1.4	3.6	1.9	20	Atraente	Feio	Atractividade
25	↑ 1.3	3.3	1.8	20	Simpático	Antipático	Atractividade
26	↑ 1.5	2.8	1.7	20	Conservador	Inovador	Inovação

<sup>138</sup> <https://www.ueq-online.org/>

Nos resultados da Tabela 7.2, nota-se que os itens relacionados à Estimulação: Aborrecido/Excitante e Motivante/Desmotivante, são os únicos itens com resultados neutros (-0,8 e 0,8). O primeiro atributo da tabela desses itens é a média da pontuação, que ficou em 0,6 pontos na escala de -3 até 3 para ambos os atributos. A segunda e a terceira coluna fornecem a variância e o desvio padrão, respectivamente, que descrevem o grau de dispersão entre as notas atribuídas. Na quarta coluna, estão os números de respostas válidas do questionário

A Figura 7.7 mostra os valores dos itens agrupados. Os itens estão agrupados nos seis conjuntos de características descritos no início do capítulo: atratividade, transparência, eficiência, controle, estimulação e inovação.

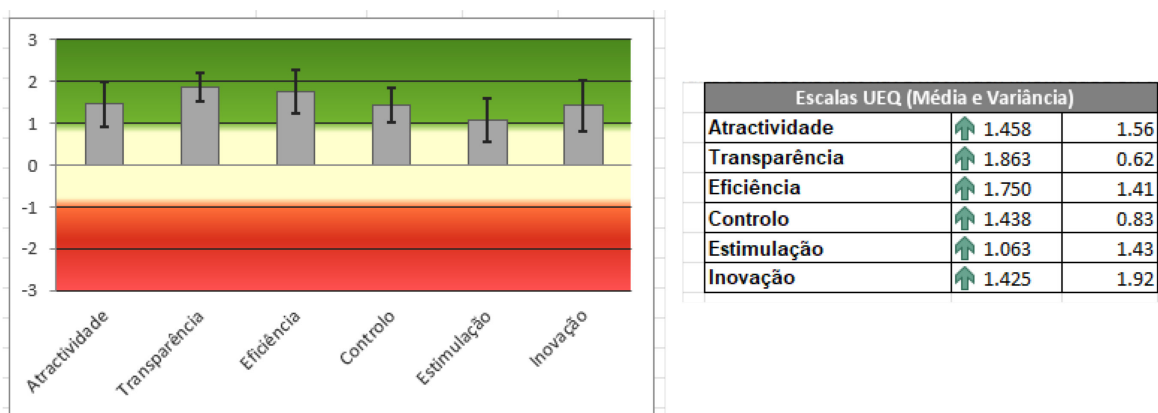


Figura 7.7: Valores por itens agrupados

Na Figura 7.7, dois quesitos destacam-se na experiência dos usuários: a transparência e a eficiência. A transparência refere-se à clareza e à compreensão das informações apresentadas no sistema bem como reflete a facilidade com que os usuários conseguem entender como o sistema funciona.

As escalas do UEQ podem ser agrupadas em qualidade pragmática (Transparência, Eficiência, Controlo) e qualidade hedônica (Estimulação, Inovação). A qualidade pragmática descreve aspectos de qualidade relacionados a tarefas, enquanto a qualidade hedônica a aspectos de qualidade não relacionados a tarefas. Abaixo da média dos três aspectos é calculada. Segundo avaliação dos usuários destaca-se a qualidade pragmática.

Na Figura 7.8, com as qualidades agrupadas, percebe-se que a qualidade pragmática destaca-se acima das outras qualidades, denotando que, entre os usuários, o produto atende às expectativas e metas práticas dos usuários, proporcionando uma experiência eficiente e eficaz

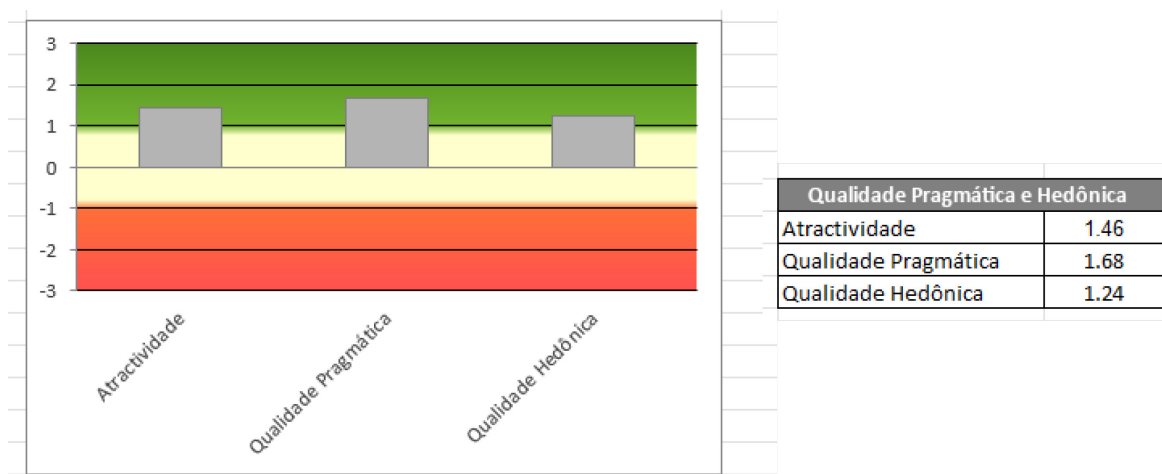


Figura 7.8: Qualidades atratividade, pragmática e hedônica.

Uma análise comparativa foi realizada com dados prévios dos pesquisadores da metodologia UEQ e podem ser visualizados na Tabela 7.3 e na Figura 7.9. As médias são definidas em relação aos valores existentes de um conjunto de referência de dados que contém 21.175 pessoas de 468 estudos sobre diferentes produtos (software comercial, aplicativos web, e lojas da web, redes sociais).

A comparação dos resultados do questionário permite conclusões sobre a qualidade relativo da aplicação web MapHidro em relação com outros produtos.

Tabela 7.3: Resultados comparados com referências de estudos prévios.

Escala	Média	Comparativo referência	Interpretação
Atratividade	1.46	Acima da média	25% de resultados melhores, 50% de resultados piores
Transparência	1.86	Bom	10% de resultados melhores, 75% de resultados piores
Eficiência	1.75	Bom	10% de resultados melhores, 75% de resultados piores
Controlo	1.44	Acima da média	25% de resultados melhores, 50% de resultados piores
Estimulação	1.06	Acima da média	25% de resultados melhores, 50% de resultados piores
Inovação	1.43	Bom	10% de resultados melhores, 75% de resultados piores

Os resultados comparativos mostraram-se positivos em todos os quesitos, notadamente transparência, eficiência e controle que obtiveram uma pontuação acima da média em relação a outros produtos testados em estudos prévios. Na Figura 7.9, pode-se ver, em gráfico de barras, a comparação em relação ao banco de dados da metodologia UEQ de estudos prévios.

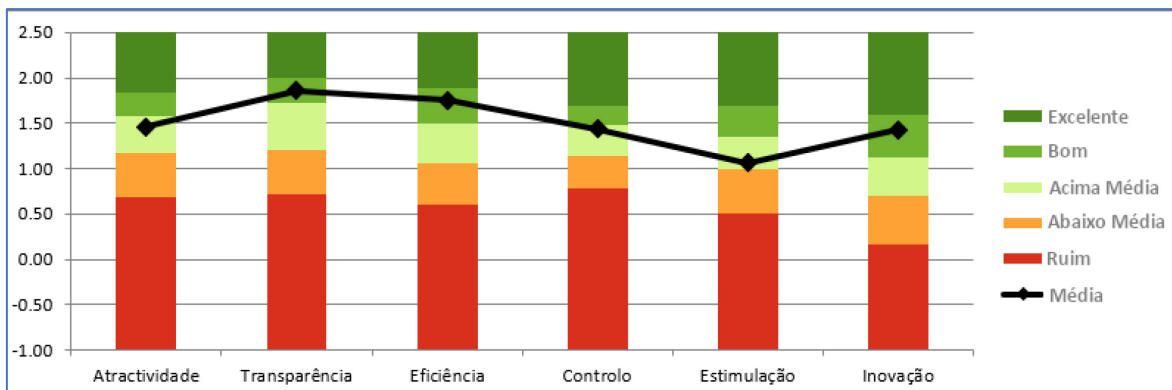


Figura 7.9: Resultados comparados com referências de estudos prévios.

A avaliação geral dos resultados do questionário de experiência de usuário mostra-se positiva em quase todos os itens individuais e agrupados. Esses resultados podem ser usados para identificar áreas de melhoria e compreender as forças e fraquezas no sistema desenvolvido e permitirão que no futuro se realizem ajustes e otimizações contínuas para aprimorar experiência na utilização da aplicação web. A escala de estimulação teve um resultado próximo de neutro, denotando que poderiam ser implementados alguns mecanismos interativos para chamar a atenção do usuário à medida que este navega pelas opções do sistema.

## Conclusões e Recomendações

O principal objetivo desta dissertação foi desenvolver uma aplicação web que apresentasse dados geocientíficos hidrológicos de maneira mais fácil para diversos perfis de profissionais e dispositivos. Na avaliação dos resultados obtidos ao longo desta dissertação entende-se que foram alcançadas as metas inicialmente traçadas nos objetivos da proposta de dissertação, nomeadamente identificar e selecionar dados de repositórios abertos de informação hidrológica; identificar, selecionar e instalar software que atendam as necessidades de processamentos das informações hidrológicas baseados em arquitetura e infraestrutura de código aberto; desenvolver uma interface fluida e responsiva; e por último, disponibilizar gráficos significativos de informação hidrológica ajustados a diferentes dispositivos.

Todo o projeto da aplicação web decorreu utilizando o processo iterativo de prototipagem em conjunto com a metodologia de *Design Science Research, Engenharia de Software*, onde há um planeamento, uma modelagem de design, seguido da construção de um protótipo, seguindo por uma implementação, feedback e comunicação com as partes relevantes/interessadas e depois repetindo todo o processo iterativamente até a sua conclusão e concretização dos objetivos esperados.

Com a modelagem das funcionalidades necessárias para atingir os objetivos, fez-se a seleção dos softwares necessários para o tratamento e processamento dos dados hidrológicos com a bibliotecas geoespaciais GMT/GDAL, apoiados pela ferramenta NodeJS projetada para construir componentes de software de rede escaláveis, com o armazenamento dos dados processados numa API apropriada configurada na camada de suporte interagindo com os outros componentes.

Através do modelo de prototipagem e comunicação permanente, a aplicação web foi construída através de consultas com potenciais especialistas na área de hidrologia de forma a deixar a aplicação mais assertiva. Através de um questionário de experiência de usuário que foi aplicado demonstrou-se que aplicação teve um impacto positivo nos usuários com

pontuação positiva em todos os quesitos (atratividade, transparência, eficiência, controle, estimulação, inovação).

## 8.1 Conclusões

Baseados nas considerações acima referidas, tratando-se do desenvolvimento de uma aplicação web com os objetivos específicos delineados no Capítulo 1, este projeto atendeu às necessidades de produção de informação diária de chuva de mais de 1500 pontos de interesse no território brasileiro, mostrando assim uma eficiência e capacidade de lidar com grandes quantidades de informação na infraestrutura informática configurada.

Os componentes desenvolvidos para interface gráfica mostraram-se adequados e adaptáveis na integração das API(s) do lado servidor e do lado do cliente para diferentes tamanhos de telas de dispositivos. As tecnologias web(*frameworks*) para a visualização através do navegador web atenderam às expectativas de transformar o aplicativo web no mais próximo que se poderia ter de uma experiência ao utilizar num programa nativo específico ao sistema operacional do dispositivo do usuário.

Nesse contexto, o trabalho desenvolvido no âmbito desta dissertação coletou informações geocientíficas públicas, que, no entanto, eram de difícil interpretação e acessibilidade, em larga medida pelo alto grau de complexidade requerido por parte do usuário para acessar, processar e interpretar a informação. Após a coleta dessa informação, através de ferramentas de código livre, o presente trabalho conseguiu transcrever e apresentar esses dados em uma informação bastante útil, de fácil interpretação e acesso, tendo sucesso num desafio segundo no qual mesmos especialistas demandariam um tempo considerável se fosse ser aplicado manualmente em cada um dos pontos de interesse pluviométrico.

Tendo atingido o principal objetivo de apresentar uma solução de código aberto em sistemas de informação que pode facilitar vida das pessoas trazendo informações mais rápidas e mais trabalhadas para entendimento mais direto da sociedade, foi ultrapassado um paradigma no contexto das geociências onde muitas dessas informações ainda estão em formatos que não podem ser lidos nem mesmo entendíveis pela mesma. Assim através do aplicativo apresentado, profissionais de geociências e gestores poderão visualizar e acessar informações de séries históricas de chuva.

## 8.2 Recomendações

Quanto a recomendações, as atividades desenvolvidas nessa dissertação encontraram diversos desafios tais como: primeiro, a falta de um conjunto de ferramentas mais integrados para descarregar, processar e apresentar os dados, fato devido por vezes à complexidade de ferramentas específicas para cada atividade. Nesse sentido novas soluções ou funcionalidades podem ser buscadas.

Segundo, as soluções com ferramentas de código livre como as apresentadas nesta dissertação já podem ser aplicadas a outras variáveis ambientais e hidrológicas que não necessariamente a estações pluviométricas ou satélites.

Por último, pensando em novas soluções ou funcionalidades, surgiram durante o desenvolvimento em paradigma de prototipagem, sugestões de funcionalidades dos especialistas em dados hidrológicos. Dada a complexidade e a limitação de tempo útil para implementação, optou-se pelo registro textual de forma que possam oportunamente serem melhor investigadas e exploradas.

Algumas dessas funcionalidades já estão, de fato, inseridas de forma experimental ou como botões na atual interface do aplicativo, mas que requerem um exame mais aprofundado em futuros trabalhos, a saber:

- funcionalidade/botão de Projetos em que o usuário registre suas estações favoritas, e outras configurações gravadas de escolhas.
- visualização de animação de mapa de calor das chuvas dos últimos dias possibilidade de visualizar períodos específicos.
- busca melhorada por filtros, como, por exemplo, onde, quando (qualquer, faixa de período, período intercalado, período de mudança) e como (somente in situ, satélite)
- aba/box/botão com a linha do tempo das últimas estações atualizadas
- aba/box/botão para selecionar e visualizar camadas de Rios, Bacias e Micro Bacias
- botão para adicionar um polígono para área de drenagem em volta da estação.
- comparar médias de precipitação de estações com outras estações.
- mapa de calor de precipitação para microbacias em volta das estações pluviométricas com os dados de satélite.
- ferramenta para selecionar uma área de polígono e calcular a precipitação para certo período desta área. (\*\*esta função exigiria processamento em tempo real e acesso a dezenas de arquivos)

- visualização de gráficos e processamento variáveis hidrológicas como altimetria entre outras. (outros repositórios e outros cálculos matemáticos).

As sugestões podem ser implementadas como os componentes de interface gráfica utilizados nesta dissertação, assim como a arquitetura pode ser adaptada para essas novas soluções encontradas.

Como palavras finais desta dissertação, espera-se que o trabalho contribua para que novos usuários e pesquisadores venham a usufruir de informações que, muitas vezes, são, em um primeiro momento, julgadas de difícil acesso, permitindo que se desenvolva uma sinergia positiva para o desenvolvimento humano e sua boa adaptação ao nosso planeta.

## Referências bibliográficas

- Benyon, D. (2019). *Designing user experience*. Pearson UK.
- Chitra, L. P., & Satapathy, R. (2017, February). Performance comparison and evaluation of Node.js and traditional web server (IIS). In *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)* (pp. 1-4). IEEE.
- Firdaus, T. (2013). *Responsive Web Design by Example Beginner's Guide*. Packt Pub.
- Frain, B. (2020). *Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques*. Packt Publishing Ltd.
- Hartson, R., & Pyla, P. S. (2018). *The UX book: Agile UX design for a quality user experience*. Morgan Kaufmann.
- Hevner, A., March, S., Park, J., and Ram, S. (2004). "Design Science in Information Systems Research." *MIS Quarterly* 28(1): 75–105.
- Hou, A. Y., Kakar, R. K., Neeck, S., Azarbarzin, A. A., Kummerow, C. D., Kojima, M., ... & Iguchi, T. (2014). The global precipitation measurement mission. *Bulletin of the American meteorological Society*, 95(5), 701-722.
- Huffman, G. J.; Adler, R. F.; Bolvin, D. T.; et al. The TRMM Multisatellite Precipitation Analysis (TMPA): Quasi-global, multiyear, combined-sensor precipitation estimates at fine scales. *Journal of Hydrometeorology*, v. 8, n. 1, p. 38–55, 2007.
- Jensen, J. R. (2014). *Introductory digital image processing: a remote sensing perspective*
- Karlsson, J. (2014). *Responsive web design with CSS frameworks*.
- Khattar, R., & Ames, D. P. (2020). A Web Services Based Water Data Sharing Approach using Open Geospatial Consortium Standards. *Open Water Journal*, 6(1), 2.
- Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings 4* (pp. 63-76). Springer Berlin Heidelberg.
- Lei, K., Ma, Y., & Tan, Z. (2014, December). Performance comparison and evaluation of web development technologies in php, python, and node. js. In *2014 IEEE 17th international conference on computational science and engineering* (pp. 661-668). IEEE.

- Lettenmaier, D. P., Alsdorf, D., Dozier, J., Huffman, G. J., Pan, M., and Wood, E. F.: Inroads of remote sensing into hydrologic science during the WRR era, *Water Resour. Res.*, 51, 7309–7342, <https://doi.org/10.1002/2015WR017616>, 2015.
- Lundmark, O. (2022). *Responsive Web Design for Modern Devices*.
- MacDonald, D. (2019). *Practical ui patterns for design systems: Fast-track interaction design for a seamless user experience*. Apress.
- Maidment, D. R. (2008). *CUAHSI hydrologic information system: overview of version 1.1*. Center for Research in Water Resources, University of Texas, Austin, TX.
- McCabe, M. F., Rodell, M., Alsdorf, D. E., Miralles, D. G., Uijlenhoet, R., Wagner, W., ... & Wood, E. F. (2017). The future of Earth observation in hydrology. *Hydrology and earth system sciences*, 21(7), 3879-3914.
- Medeiros, Marcelo (2023). *Gestão da Rede Hidrometeorológica Nacional [PowerPoint slides]*. Agência Nacional de Águas e Saneamento Básico. <https://www.gov.br/censipam/pt-br/eventos/ISHA2023Dia27MesaRedonda1MarceloMedeirosANADivulgao.pdf>
- Norman, D., & Nielsen, J. (2016). *The definition of user experience*. NN/g Nielsen Norman Group. <https://www.nngroup.com/articles/definition-user-experience/>, (accessed 2020-03-20).
- Prakash, P., Biju, R., & Kamath, M. (2015, January). Performance analysis of process driven and event driven web servers. In *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)* (pp. 1-7). IEEE.
- Rauschenberger, M., Schrepp, M., Pérez Cota, M., Olschner, S., & Thomaschewski, J. (2013). Efficient measurement of the user experience of interactive products. How to use the user experience questionnaire (UEQ). Example: Spanish language version.
- Roger, S. P., & Bruce, R. M. (2015). *Software engineering: a practitioner's approach*. McGraw-Hill Education.
- Rozante, J. R., Moreira, D. S., de Goncalves, L. G. G., & Vila, D. A. (2010). Combining TRMM and surface observations of precipitation: technique and validation over South America. *Weather and forecasting*, 25(3), 885-894
- Saks, E. (2019). *JavaScript Frameworks: Angular vs React vs Vue*.
- Scott Jr, E. A. (2015). *SPA Design and Architecture: Understanding single-page web applications*. Simon and Schuster.
- Sood, A., and Smakhtin, V. (2015). Global hydrological models: a review. *Hydrol. Sci. J.* 60, 549–565. doi: 10.1080/02626667.2014.950580
- Sommerville, I. (2016). *Engenharia de software*. 9a. edição, Addison-Wesley/Pearson.
- Stackoverflow, (2022). *Most popular technologies*

<https://survey.stackoverflow.co/2022#technology-most-popular-technologies>.

Statcounter, (2023). Desktop vs Mobile vs Tablet Market Share Worldwide.  
<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide>

Steiniger, S., Weibel, R., & Warf, B. (2010). GIS software: a description in 1000 words

Sun, Q., Miao, C., Duan, Q., Ashouri, H., Sorooshian, S., & Hsu, K. L. (2018). A review of global precipitation data sets: Data sources, estimation, and intercomparisons. *Reviews of Geophysics*, 56(1), 79-107.

Tsou, M. H. (2011). Revisiting web cartography in the United States: The rise of user-centered design. *Cartography and geographic information science*, 38(3), 250-257.

Vaishnavi, V. K., & Kuechler, W. (2015). Design science research methods and patterns: Innovating information and communication technology. *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*.  
<https://doi.org/10.1201/9781420059335>

Zunino, A., Velázquez, G., Celemín, J. P., Mateos, C., Hirsch, M., & Rodriguez, J. M. (2020). Evaluating the Performance of Three Popular Web Mapping Libraries: A Case Study Using Argentina's Life Quality Index. *ISPRS International Journal of Geo-Information*, 9(10), 563.

Wohlgethan, E. (2018). Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue.js (Doctoral dissertation, Hochschule für Angewandte Wissenschaften Hamburg).

## Anexo I - Questionário de Experiência de Usuário

Para avaliar a experiência de Usuário ao navegar pelo aplicativo MapHidro foi utilizada a metodologia UEQ (Laugwitz, B, et al.,2008). Esta técnica permite uma avaliação rápida de qualquer produto interativo. As escalas do questionário são projetadas para cobrir uma impressão abrangente de experiência de usuário. O formato do questionário permite ao usuário expressar imediatamente sentimentos, impressões e atitudes que surgem quando eles usam um produto.

24(Vinte e quatro) pessoas no total responderam ao questionário inserido no aplicativo. Os resultados podem ser vistos no capítulo 7. O questionário de experiência de usuário contém 6 escalas com 26 itens no total, e está disponível para visualização no apêndice I desta dissertação. As escalas estão alinhadas da seguinte forma:

<b>1</b>	<b>Atratividade:</b> Impressão geral em relação ao produto. Os usuários gostam ou desgostam do produto. Itens: Agradável/Desagradável; Bom/Mau; Desinteressante/Atrativo; Incômodo/Cômodo; Atraente/Feio; Simpático/Antipático;
<b>2</b>	<b>Transparência:</b> É fácil entender como usar o produto? É fácil se familiarizar com o produto? Itens: Incompreensível/Compreensível; De Fácil aprendizagem/De difícil aprendizagem; Complicado/Fácil; Evidente/Confuso;
<b>3</b>	<b>Eficiência:</b> Ele é um produto rápido e eficiente? A interface é organizada? Itens: Rápido/Lento; Ineficiente/Eficiente; Impraticável/Prático; Organizado/Desorganizado;
<b>4</b>	<b>Controle:</b> O usuário se sente em controle da interação? É a interação previsível? Itens: imprevisível/previsível; obstrutivo/conductor; seguro/ inseguro;
<b>5</b>	<b>Estimulação:</b> É interessante e excitante usar o produto? O usuário sente-se motivado para avançar no uso do produto? Itens: valioso/sem valor; aborrecido/excitante; desinteressante/interessante; motivante/desmotivante;
<b>6</b>	<b>Inovação:</b> É o design do produto inovativo e criativo? O produto chama a atenção do usuário? Itens: criativo/sem criatividade; original/convencional; comum/Vanguardista; conservador/Inovador

### Questionário de Experiência de Usuário

Por favor, dê-nos a sua opinião.

A fim de avaliar o produto, por favor preencha o seguinte questionário. É constituído por pares de opostos relativos às propriedades que o produto possa ter. As graduações entre os opostos são representadas por círculos. Ao marcar um dos círculos, você pode expressar sua opinião sobre um conceito.

**Exemplo:**

<b>Atraente</b>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Feio</b>
-----------------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-------------

Esta resposta significa que avalia o produto mais atraente do que feio.

Marque a sua resposta da forma mais espontânea possível. É importante que não pense demasiado na resposta porque a sua avaliação imediata é que é importante.

Por favor, assinale sempre uma resposta, mesmo que não tenha certezas sobre um par de termos ou que os termos não se enquadrem com o produto.

Não há respostas "certas" ou respostas "erradas". A sua opinião pessoal é que conta!

Por favor, dê-nos a sua avaliação atual do produto em causa.

Por favor, marque apenas um círculo por linha

	1	2	3	4	5	6	7	
<b>Desagradável</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Agradável</b>
<b>Incompreensível</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Compreensível</b>
<b>Criativo</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Sem criatividade</b>
<b>De Fácil aprendizagem</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>De difícil aprendizagem</b>
<b>Valioso</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Sem valor</b>
<b>Aborrecido</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Excitante</b>
<b>Desinteressante</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Interessante</b>
<b>Imprevisível</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Previsível</b>
<b>Rápido</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Lento</b>
<b>Original</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Convencional</b>
<b>Obstrutivo</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Condutor</b>
<b>Bom</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Mau</b>
<b>Complicado</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Fácil</b>
<b>Desinteressante</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Atrativo</b>
<b>Comum</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Vanguardista</b>
<b>Incómodo</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Cómodo</b>
<b>Seguro</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Inseguro</b>
<b>Motivante</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Desmotivante</b>
<b>Atende as expectativas</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Não atende as expectativas</b>
<b>Ineficiente</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Eficiente</b>
<b>Evidente</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Confuso</b>
<b>Impraticável</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Prático</b>
<b>Organizado</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Desorganizado</b>
<b>Atraente</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Feio</b>
<b>Simpático</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Antipático</b>
<b>Conservador</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<b>Inovador</b>