

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Desenvolvimento da BikeScience/OD Web

Thiago Jose Benitez Pena

MONOGRAFIA FINAL

MAC 499 — TRABALHO DE
FORMATURA SUPERVISIONADO

Supervisor: Prof. Dr. Higor Amario de Souza
Cossupervisor: Prof. Dr. Fabio Kon
Cossupervisora: Dra. Letícia Lindenberg Lemos

São Paulo
2021

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)*

Agradecimentos

À minha mãe por todo o suporte e carinho ao longo da minha vida. À minha esposa Vivi pela paciência e compreensão durante 15 anos e contando. Aos orientadores Higor Amario de Souza, Letícia Lindenberg Lemos e Fabio Kon pela paciência, direcionamento e conversas durante o ano.

Resumo

Thiago Jose Benitez Pena. **Desenvolvimento da BikeScience/OD Web**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

Este trabalho descreve o desenvolvimento e implementação da BikeScienceWeb, uma aplicação web interativa que busca facilitar o acesso às funcionalidades da BikeScience, uma ferramenta de ciência de dados de código aberto que usa dados de sistemas de compartilhamento de bicicletas e pesquisas de viagens para analisar os padrões no uso das bicicletas, gerando assim conhecimento que pode ser usado para apoiar o planejamento futuro e melhorias em relação ao ciclismo urbano. O sistema de software foi desenvolvido em Python, Pandas e GeoPandas, em conjunto com o arcabouço Flask, no *back-end*, e sua interface web em JavaScript utilizando a biblioteca de mapas Leaflet e o arcabouço de interfaces gráficas Vue.js. Para avaliação da aceitação da ferramenta realizamos uma pesquisa com o Modelo de Aceitação Tecnológica (TAM). Entre os participantes foram escolhidos funcionários da CET-SP envolvidos com a BikeScience e pessoas ligadas à mobilidade urbana. Os participantes avaliaram bem a BikeScienceWeb, indicando intenção de adotá-la para suas análises de mobilidade urbana. Por outro lado, em relação à utilidade e facilidade de uso, apesar de positivas em geral, as avaliações indicam possibilidade de melhorias, mas sem impedir seu uso por parte dos funcionários da CET-SP.

Palavras-chave: Mobilidade. Ciência de dados. Visualização. Software de código aberto. Engenharia de *software*. Sistema de *software*. Aplicação *web*.

Abstract

Thiago Jose Benitez Pena. **The Development of BikeScience/OD Web**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2021.

This monograph describes the development and implementation of BikeScienceWeb, an interactive web application that seeks to facilitate access to the features of BikeScience, an open-source data science tool that uses data from bike-sharing systems and travel surveys to analyze patterns in the use of bicycles, thus generating knowledge that can be used to support future planning and improvements in urban cycling. The software system was developed in Python, Pandas, and GeoPandas, within the Flask framework, in the back-end. Its web interface was developed in JavaScript using the Leaflet map library and the Vue.js graphical user interface framework. To assess the acceptance of the tool, we conducted a survey using the Technology Acceptance Model (TAM). Among the participants, we chose CET-SP employees involved with BikeScience and people linked to urban mobility. Participants rated BikeScienceWeb well, showing their intention to adopt it for their urban mobility analyses. On the other hand, in relation to usefulness and ease of use, despite being positive in general, the evaluations indicate the possibility of improvements, but without restraining its use by CET-SP employees.

Keywords: Mobility. Data science. Visualization. Open source software. Software engineering. Software systema. Web application.

Lista de Abreviaturas

API	Interface de Programação de Aplicações (<i>Application Programming Interface</i>)
CET-SP	Companhia de Engenharia de Tráfego de São Paulo
CSV	Valores separados por vírgulas (<i>Comma-separated values</i>)
IDE	Ambiente de Desenvolvimento Integrado (<i>Integrated Development Environment</i>)
IME	Instituto de Matemática e Estatística
INCT	Instituto Nacional de Ciência e Tecnologia
Pesquisa OD	Pesquisa Origem Destino do Metrô
RMSP	Região Metropolitana de São Paulo
SaaS	Software como um Serviço
SPA	Aplicativo de Página Única (<i>Single Page Application</i>)
SSH	<i>Secure Shell</i>
UI	Interface do Usuário <i>User Interface</i>
UML	Linguagem de Modelagem Unificada (<i>Unified Modeling Language</i>)
URL	Localizador Uniforme de Recursos (<i>Uniform Resource Locator</i>)
USP	Universidade de São Paulo

Lista de Figuras

3.1	A interface do usuário da BikeScienceWeb	8
3.2	Zonas OD 2017 e os limites dos municípios da RMSP Fonte: OD (2017d) .	10
3.3	Composição das viagens segundo a Pesquisa OD 2017 Fonte: OD (2019) .	11
3.4	Zonas OD na BikeScienceWeb	14
3.5	Parâmetros de deslocamento da grade no BikeScienceWeb	14
3.6	Um exemplo de fluxo no BikeScienceWeb	15
3.7	Um exemplo de cálculo de fluxo	19
3.8	Aba do seletor de fluxos na UI	21
3.9	A edição da grade na UI	21
3.10	Aba do seletor de filtros na UI	22
3.11	Aba do seletor de camadas na UI	23
3.12	Estrutura genérica de um software com Vue.js e Vuex Fonte: VUEX (2021)	25
3.13	Rede interna do BikeScienceWeb	27
3.14	Diagrama de contexto para a BikeScienceWeb	30
3.15	Diagrama de contêineres para a BikeScienceWeb	31
3.16	Diagrama de componentes do Front-end para a BikeScienceWeb	32
3.17	Diagrama de componentes do Back-end para a BikeScienceWeb	33
3.18	Diagrama do classes Controlador de Filtros	35
3.19	Diagrama de classes do Controlador de Camadas	36
3.20	Diagrama de classes do Controlador de Grade	36
3.21	Quantidade excessiva de fluxos em São Paulo	37
3.22	Quantidade excessiva de fluxos em Cerqueira César	38
3.23	Fluxos em Cerqueira César	38
3.24	Fluxos em Cerqueira César com camadas de metrô e de infraestrutura ciclovária	39
4.1	Respostas às perguntas da avaliação TAM.	43
4.2	Áreas de atuação dos participantes.	43
4.3	Anos de experiência dos participantes na área de atuação.	44

4.4	Trabalho ou atuação dos participantes.	44
4.5	Gênero dos participantes.	45

Lista de Tabelas

3.1	Modais avaliados na Pesquisa OD 2017. Fonte: OD (2017a)	11
3.2	Campos de localização multimodais da Pesquisa OD 2017 Fonte: OD (2017a)	12
3.3	Filtros da Pesquisa OD utilizados na BikeScienceWeb Fonte: OD (2017a)	16
3.4	Filtros e suas categorias na BikeScienceWeb Fonte: OD (2017a)	17
3.5	Camadas utilizadas na BikeScienceWeb	18
4.1	Perguntas do formulário de pesquisa de avaliação da BikeScienceWeb	42
4.2	Trabalho ou atuação dos participantes	45
4.3	Respostas dos participantes às perguntas P4 e P5	46

Lista de Programas

3.1	Algoritmo 1: Cálculo dos fluxos.	20
3.2	Algoritmo 2: Cálculo dos centros de massa.	20

Sumário

1	Introdução	1
1.1	Objetivos	1
1.2	Motivações	1
1.3	Metodologia	2
1.4	Organização	2
2	Contextualização	3
2.1	Projeto InterSCity	3
2.2	BikeScience	4
2.2.1	A BikeScience e a CET	4
3	BikeScienceWeb	7
3.1	Pesquisa Origem Destino	8
3.1.1	Zonas Origem e Destino	9
3.1.2	Fator de expansão da viagem	9
3.1.3	Modais	10
3.1.4	Viagens multimodais	12
3.1.5	Extração e tratamento dos dados das viagens	12
3.2	Áreas de segregação	13
3.3	Fluxos	13
3.4	Filtros de viagens	16
3.5	Camadas	17
3.6	Cálculo dos Fluxos	18
3.7	Interface do usuário	20
3.8	Tecnologias utilizadas	22
3.8.1	Python	23
3.8.2	Flask	23
3.8.3	Pandas	23

3.8.4	GeoPandas	24
3.8.5	Vue.js	24
3.8.6	Vuex	24
3.8.7	Leaflet e Vue2Leaflet	25
3.8.8	Docker e Docker Compose	26
3.8.9	NGINX	26
3.8.10	HTML, CSS e JavaScript	27
3.8.11	JSON	27
3.8.12	NPM	27
3.8.13	Buefy	28
3.8.14	Toastr e Vue-toastr-2	28
3.8.15	ESLint	28
3.9	Arquitetura da BikeScienceWeb	28
3.9.1	O modelo C4	28
3.10	Ambiente de produção	37
3.11	Exemplo de aplicação	37
4	Avaliação	41
4.1	Metodologia	41
4.2	Questionário	41
4.3	Resultados	42
4.4	Discussão dos resultados	45
5	Conclusão e Trabalhos Futuros	49
	Referências	51

Capítulo 1

Introdução

1.1 Objetivos

A BikeScience¹ é uma ferramenta que possibilita que gestores públicos, empresas envolvidas, pesquisadores e entusiastas em mobilidade urbana analisem e entendam o impacto do ciclismo urbano nas cidades. Este trabalho tem como objetivo implementar uma versão web para a BikeScience, a BikeScienceWeb², para facilitar a disseminação de seu uso por parte de especialistas em mobilidade.

1.2 Motivações

A BikeScience utiliza dados sobre pesquisas de viagens, como a Pesquisa Origem Destino, e também dados de sistemas de compartilhamento de bicicletas para gerar informação sobre os padrões de uso das bicicletas através de mapas e gráficos. Desenvolvida em python, a BikeScience usa pandas e geopandas para a análise de dados.

O uso da ferramenta pela Companhia de Engenharia de Tráfego de São Paulo (CET-SP) mostrou dificuldades para sua adoção, por exigir conhecimentos de programação em Python e Pandas, além de familiaridade no uso do Jupyter Notebook. Outra dificuldade enfrentada era a necessidade de instalação de bibliotecas e arquivos nas máquinas para utilizar a ferramenta. Máquinas defasadas com ambientes Windows, a necessidade de autorização para instalação como administrador e atualizações que exigiam repetir todo esse processo. Na prática esses fatores inviabilizavam seu uso no longo prazo. A BikeScienceWeb busca eliminar essas necessidades e dificuldades fornecendo as análises presentes na BikeScience em uma versão *web* interativa.

¹ <https://gitlab.com/intercity/bike-science>

² <http://bikescienceweb.intercity.org/>

1.3 Metodologia

Para avaliação da ferramenta BikeScienceWeb foi aplicado um questionário baseado nas questões do Modelo de Aceitação Tecnologia (TAM), modelo que avalia como os usuários percebem a utilidade, a facilidade de uso e a intenção de uso de uma tecnologia. Foram selecionados especialistas em mobilidade urbana, incluindo funcionários da CET-SP e pessoas ligadas à mobilidade urbana.

A maioria das avaliações da BikeScienceWeb foi positiva nas três categorias da TAM, indicando intenção de adoção da ferramenta pelos participantes em suas análises de mobilidade urbana.

1.4 Organização

Este trabalho se divide da seguinte forma: no Capítulo 2 apresentamos o projeto InterSCity e a ferramenta BikeScience nele desenvolvida. Discutimos também a parceria com a CET-SP e sua importância para o projeto.

No Capítulo 3 apresentamos a Pesquisa Origem Destino do Metrô (Pesquisa OD), sua relevância para a BikeScience e a BikeScienceWeb e como essas ferramentas utilizam seus dados. Apresentamos também os conceitos de áreas de segregação, fluxos de viagens, que são definições fundamentais para a aplicação do modelo. Em seguida, descrevemos as funcionalidades, o cálculo dos fluxos, a interface *web*, as tecnologias utilizadas, a arquitetura, o ambiente de produção e um exemplo de aplicação.

O Capítulo 4 descreve em detalhes a metodologia utilizada na avaliação da ferramenta e a discussão dos resultados. Finalmente, no Capítulo 5 discutimos as conclusões e trabalhos futuros da BikeScienceWeb.

Capítulo 2

Contextualização

2.1 Projeto InterSCity

O projeto InterSCity¹ é um projeto de pesquisa colaborativa criado pelo Instituto Nacional de Ciência e Tecnologia (INCT) da Internet do Futuro para Cidades Inteligentes, sediado no Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP) e financiado pela CNPq, CAPES e FAPESP. O Projeto inclui nove universidades brasileiras, além de parceiros nacionais e internacionais (BATISTA *et al.*, 2016).

De forma geral, o objetivo do InterSCity é realizar pesquisas para contribuir com conhecimento sobre cidades inteligentes. O projeto busca desenvolver plataformas de código aberto integradas contendo os alicerces para o desenvolvimento de aplicações robustas, integradas e sofisticadas para as cidades inteligentes do futuro (INTERSCITY, 2021).

Os desafios e iniciativas do projeto são organizadas, segundo BATISTA *et al.*, 2016 em três frentes:

1. Redes e computação distribuídas de alto desempenho;
2. Engenharia de software para a internet do futuro;
3. Análise e modelagem matemáticas para a internet do futuro e cidades inteligentes.

BATISTA *et al.* (2016) também definem os principais objetivos do projeto dentre os quais citamos os dois que melhor se alinham com as iniciativas descritas neste trabalho:

- (1) O desenvolvimento de modelos matemáticos analíticos para a representação, estudo e análise de grandes conjuntos estruturados de objetos em ambientes de cidades inteligentes do futuro. Esses modelos serão fundamentais para a gestão e planejamento eficientes do uso e implantação de recursos em cidades inteligentes, com o objetivo de otimizar o desempenho, maximizar a sustentabilidade de sistemas e promover economia de energia e de outros recursos.

¹ <https://interscity.org>

- (6) Aplicação de tecnologias desenvolvidas no projeto em cenários do mundo real que envolvam serviços reais de cidades inteligentes, com o objetivo de melhorar a qualidade de vida dos cidadãos e de fornecer soluções para uma gestão mais eficiente de grandes e megacidades. Isso será alcançado tanto por meio de parcerias com governos municipais quanto por transferência de tecnologia para os setores público e privado, considerando especialmente *startups*.

Na próxima subseção discutiremos como a BikeScience se insere no contexto do InterSCity.

2.2 BikeScience

A BikeScience é uma ferramenta de ciência de dados de código aberto para a análise e planejamento de mobilidade ativa em ambientes urbanos. Foi criada em um projeto colaborativo por pesquisadores do projeto InterSCity da Universidade de São Paulo e pelo MIT Senseable City Lab² (KON *et al.*, 2020). A BikeScience é desenvolvida na linguagem Python em conjunto com bibliotecas avançadas de análise de dados, visualização e geolocalização, como Pandas, GeoPandas, Numpy, Scipy, Matplotlib e Folium.

A ferramenta usa dados de sistemas de compartilhamento de bicicletas e pesquisas de viagens para analisar os padrões no uso das bicicletas, gerando assim conhecimento que pode ser usado para apoiar o planejamento futuro e melhorias em relação ao ciclismo urbano.

Entre as análises já existentes no BikeScience, citamos as seguintes:

- mapas mostrando os principais fluxos de viagens de bicicleta em regiões de interesse;
- sugestões de rotas dadas uma zona de origem e uma zona de destino;
- estatísticas sobre viagens de bicicleta, perfil dos ciclistas e locais onde as viagens ocorrem;
- as zonas da cidade com maior concentração de viagens de bicicletas;
- e mapas de calor mostrando as estações de bicicletas mais utilizadas.

2.2.1 A BikeScience e a CET

A CET-SP e o Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP) iniciaram um convênio para o uso de técnicas e ferramentas de pesquisa aplicada em Cidades Inteligentes para a solução de problemas reais relacionados à mobilidade em São Paulo (SOUZA *et al.*, 2021).

A partir dessa parceria, a CET-SP está utilizando a ferramenta BikeScience com o objetivo de atender a sua necessidade de dados sobre a circulação de bicicletas para o planejamento de expansão e gestão do sistema ciclovitário da cidade de São Paulo.

² <https://senseable.mit.edu>

A BikeScience pode ser usada para analisar e entender o impacto do ciclismo urbano nas cidades e, dessa forma, auxiliar os gestores públicos no planejamento e na avaliação de locais que precisam de maior atenção com relação à implantação e manutenção da infraestrutura cicloviária (SOUZA *et al.*, 2021).

Uma das principais funcionalidades da BikeScience é a geração de fluxos de viagens mais significativos entre áreas de uma cidade, sendo justamente essa a funcionalidade base da BikeScienceWeb neste presente trabalho.

Nesse sentido, nota-se grande correlação com os objetivos 1 e 6 do InterSCity citados anteriormente. Com o objetivo 1, por tratar-se de um modelo matemático para representar viagens de bicicleta, servindo de base para a gestão e planejamento de implantação de recursos de infraestrutura cicloviária. Já em relação ao objetivo 6, a ferramenta tem como objetivo uma gestão mais eficiente da cidade, melhorando a qualidade de vida dos cidadãos.

Algumas das dificuldades encontradas no uso da BikeScience pela CET-SP são que seu uso requer a instalação de dependências, conhecimentos de programação em Python e Pandas, familiaridade de uso do Jupyter Notebook, além de eventuais atualizações necessárias.

Sem esse conhecimento, a utilização da ferramenta estava bastante limitada, restringindo a possibilidade de alterar parâmetros para realizar novas análises. O uso de diferentes sistemas operacionais também trás a dificuldade de se garantir o uso das mesmas versões das dependências instaladas em diferentes máquinas. Nesse contexto, para tratar dessas dificuldades foi criada a BikeScienceWeb, que será discutida no Capítulo 3.

Capítulo 3

BikeScienceWeb

A BikeScienceWeb¹ busca transformar as análises da ferramenta BikeScience em uma aplicação *web* de código aberto². Ela foi criada para possibilitar que gestores públicos e outros possíveis interessados possam fazer suas análises sem necessitar dos conhecimentos de programação exigidos pela BikeScience. Além disso, uma vantagem de aplicações *web* é a independência de sistema operacional e hardware.

Aplicações *web* são softwares acessíveis via navegador de internet. TANEJA e GUPTA, 2014 citam algumas vantagens de aplicações *web* se comparadas às aplicações *desktop*, citadas a seguir:

- Não geram custos de publicação e distribuição;
- Não precisam ser instalados na máquina do usuário;
- Podem ser acessadas por um grande número de usuários;
- Backups e atualizações são realizados com facilidade;
- São independentes do dispositivo, do sistema operacional e do local de acesso.

A BikeScienceWeb foi modelada como um Aplicativo de Página Única, ou *Single Page Application* (SPA). Na definição de JADHAV *et al.* (2015), um SPA é composto por componentes individuais que podem ser substituídas/atualizadas independentemente, sem a necessidade de recarregar a página inteira a cada ação do usuário.

Segundo CHOI, 2020, o objetivo de usar aplicações do tipo SPA é fazer com que a aplicação passe a sensação de ser uma aplicação nativa. Aplicações *web* tradicionais costumam ter lentidão e qualquer mudança na página exige uma chamada ao servidor para obter uma nova tela. Aplicações do tipo SPA renderizam imediatamente apenas as partes da tela sem precisar esperar um novo arquivo vindo do servidor.

Para diminuir as esperas por uma resposta do servidor, aplicações do tipo SPA costumam baixar inicialmente os arquivos necessários para sua execução básica, e depois baixam o

¹ <http://bikescienceweb.interscity.org/>

² <https://gitlab.com/interscity/bike-science-web>

restante de forma assíncrona sob demanda, para minimizar a espera.

A Figura 3.1 mostra a interface do usuário da BikeScienceWeb diretamente no navegador. Para o exemplo da figura, ativamos algumas camadas, um filtro e selecionamos os fluxos dos quartis 1 e 2 para serem exibidos no mapa. A interface do usuário será detalhada na Seção 3.7.

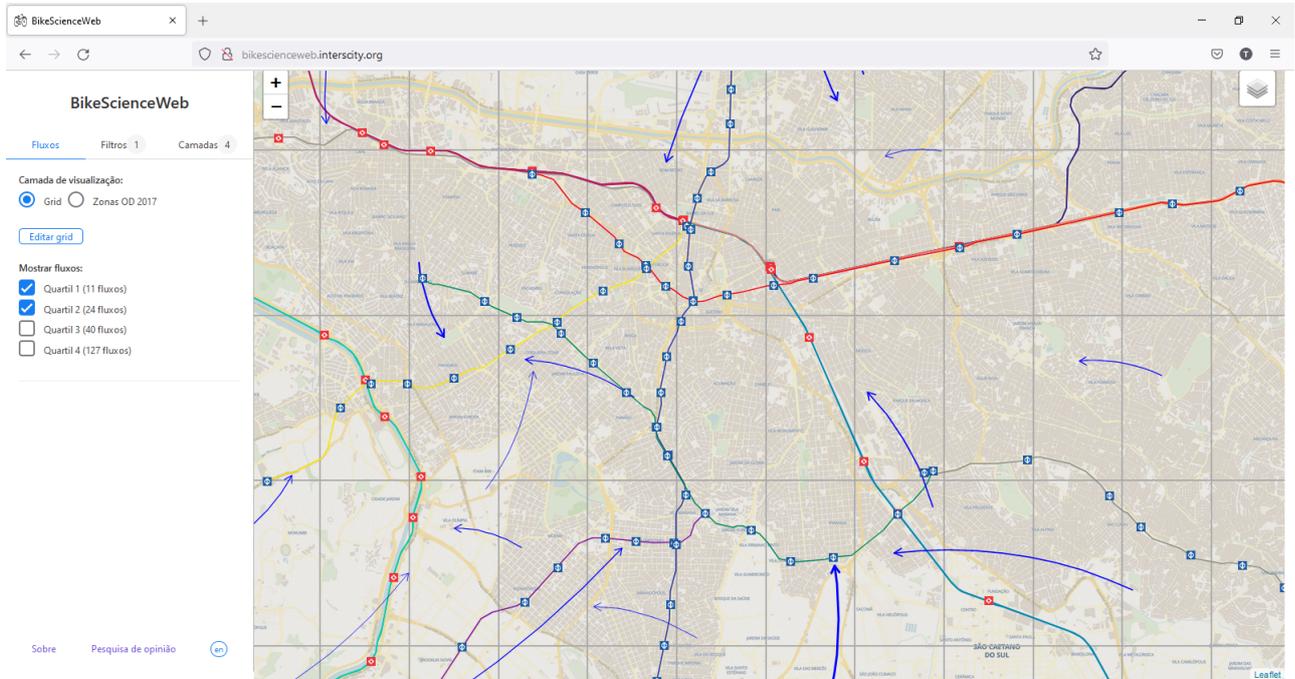


Figura 3.1: A interface do usuário da BikeScienceWeb

3.1 Pesquisa Origem Destino

Nesta seção, contextualizamos a Pesquisa OD com as ferramentas BikeScience e BikeScienceWeb. Os dados provenientes da pesquisa são a base de dados utilizada para a aplicação neste trabalho, embora a ferramenta não se limite a ela, podendo utilizar outras fontes de dados para novas análises. Nesse contexto, esta seção apresenta a definição das zonas de origem e destino que são fundamentais para a definição e cálculo dos fluxos. Em seguida, apresentamos outros conceitos importantes presentes nos dados da pesquisa, como fator de expansão da viagem, modais e viagens multimodais. Finalmente, é mostrado brevemente o processo de tratamento dos dados para que possam ser utilizados pela ferramenta.

A Pesquisa Origem e Destino (Pesquisa OD) é uma pesquisa sobre os padrões de mobilidade que as pessoas fazem diariamente na Região Metropolitana de São Paulo (RMSP). Além da informação de origens e destinos, a pesquisa levanta também os motivos, os modos de transporte dessas viagens, além de dados socioeconômicos da população, (OD, 2019).

A pesquisa é realizada pela Companhia do Metrô de São Paulo, com apoio da Secretaria dos Transportes Metropolitanos do Estado de São Paulo, à qual é subordinada, e a

participação de diversas outras instituições abrangendo a Região Metropolitana de São Paulo a cada dez anos, desde 1967 (OD, 2017b)).

A Pesquisa OD 2017 tem três objetivos principais:

- quantificar e caracterizar o padrão atual de viagens na Região Metropolitana de São Paulo;
- obter insumos para projeção de viagens futuras;
- obter subsídios para planos e projetos futuros.

Os resultados da Pesquisa OD são essenciais para os planejadores urbanos, pois mostram os fluxos dos cidadãos entre as regiões, apontando o impacto das escolhas de transporte na conformação das cidades e de seus bairros, fornecendo subsídio importante para os gestores elaborarem seus planos urbanos e seus planos de mobilidade, (OD, 2017c).

A Pesquisa OD fornece dados de insumo para as diversas análises feitas na BikeScience e BikeScienceWeb. Entretanto, SOUZA *et al.*, 2021 destaca que a BikeScience não se limita a ela, utilizando também outras fontes de dados de sistemas de compartilhamento de bicicletas, como o BikeSampa e a Yellow. Além dessas, é possível utilizar também outras fontes de dados, desde que sejam tratados para se adequar ao modelo utilizado nas análises.

3.1.1 Zonas Origem e Destino

Antes mesmo da coleta dos dados para a pesquisa são definidas as zonas de origem e destino, que são áreas com condições urbanas similares que servem como base para o dimensionamento da amostra. Essas zonas não coincidem necessariamente com os limites de municípios e distritos e podem variar com o tempo, dependendo, dentre outros fatores, do crescimento das áreas urbanizadas.

De acordo com OD (2017d)

Os critérios para definição do zoneamento de 2017 foram os seguintes: área compatível com o zoneamento da Pesquisa Origem e Destino 2007; compatível com os limites de municípios e distritos no Município de São Paulo; levando em conta os limites dos setores censitários do IBGE.

Foram ainda considerados no zoneamento o sistema de transporte, os equipamentos urbanos, as barreiras físicas, as áreas vazias, o crescimento de áreas urbanizadas, novos equipamentos urbanos e de transporte em planejamento ou projeto.

Na Figura 3.2 mostramos como ficaram os limites das Zonas OD 2017 e como eles diferem dos limites dos municípios. Essas zonas são importantes pois serão usadas para o cálculo dos fluxos, como será mostrado mais adiante.

3.1.2 Fator de expansão da viagem

O fator de expansão da viagem é um número que determina o valor que uma viagem da amostra da pesquisa representa no total, ou seja, cada viagem da amostra representa uma quantidade FE_VIA de viagens da população.

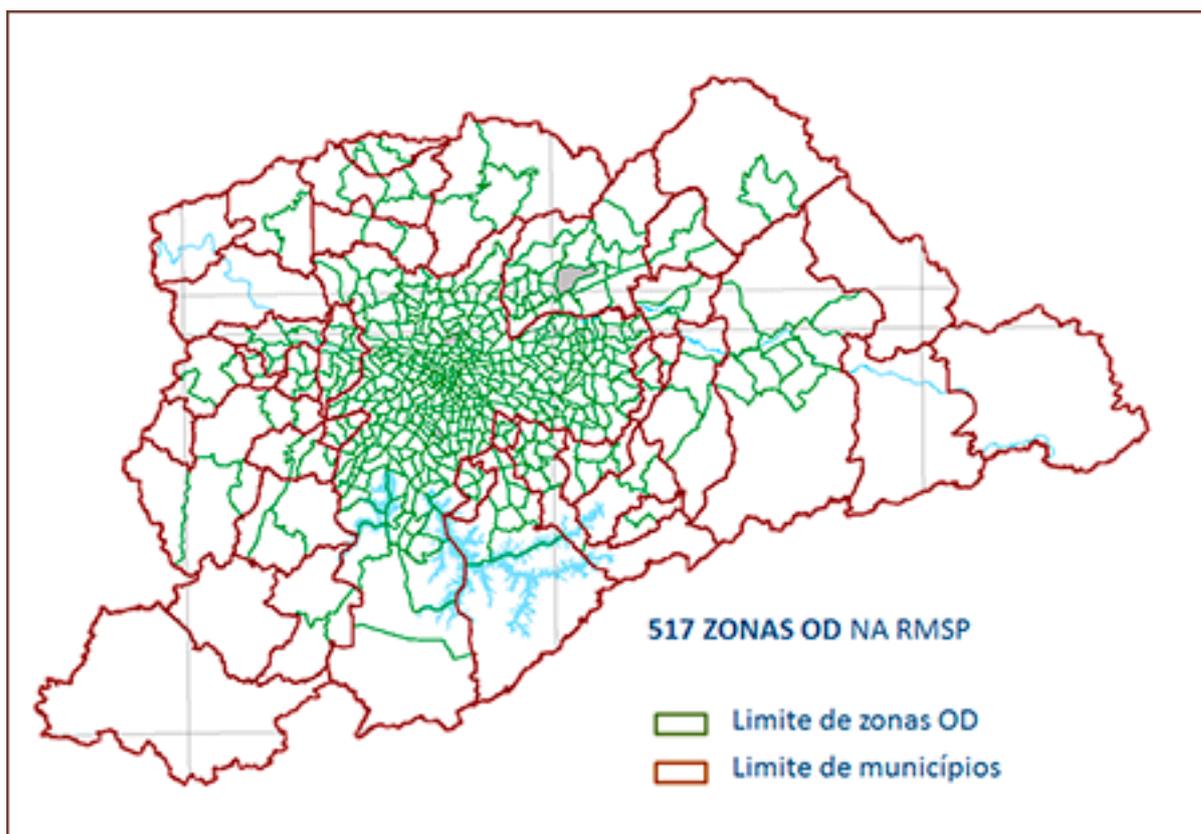


Figura 3.2: Zonas OD 2017 e os limites dos municípios da RMSP
Fonte: OD (2017d)

O cálculo do fator de expansão da amostra para o universo de cada um dos segmentos foi obtido conforme indicado (OD, 2019):

- **Rodovias:** contagem volumétrica classificada de veículos realizada durante a pesquisa.
- **Aeroportos:** dados fornecidos pela Infraero – administradora do aeroporto de Congonhas – e GRU Airport – concessionária do aeroporto de Guarulhos.
- **Terminais rodoviários:** dados fornecidos pela Socicam.
- **Ônibus fretados:** contagem de embarques realizada durante a pesquisa.

3.1.3 Modais

A Pesquisa OD 2017 analisou viagens em diversos modais de transporte, apresentados na Tabela 3.1 em conjunto com o índice que o representa na base de dados.

A Figura 3.3 apresenta a composição das viagens por tipo de modal resultante da Pesquisa OD 2017 e a comparação com os valores da Pesquisa OD 2007.

Código	Modal
01	Metrô
02	Trem
03	Monotrilho
04	Ônibus/micro-ônibus/perua do município de São Paulo
05	Ônibus/micro-ônibus/perua de outros municípios
06	Ônibus/micro-ônibus/perua metropolitano
07	Transporte Fretado
08	Transporte Escolar
09	Dirigindo Automóvel
10	Passageiro de Automóvel
11	Táxi Convencional
12	Táxi não Convencional
13	Dirigindo Moto
14	Passageiro de Moto
15	Bicicleta
16	A Pé
17	Outros

Tabela 3.1: Modais avaliados na Pesquisa OD 2017.

Fonte: OD (2017a)

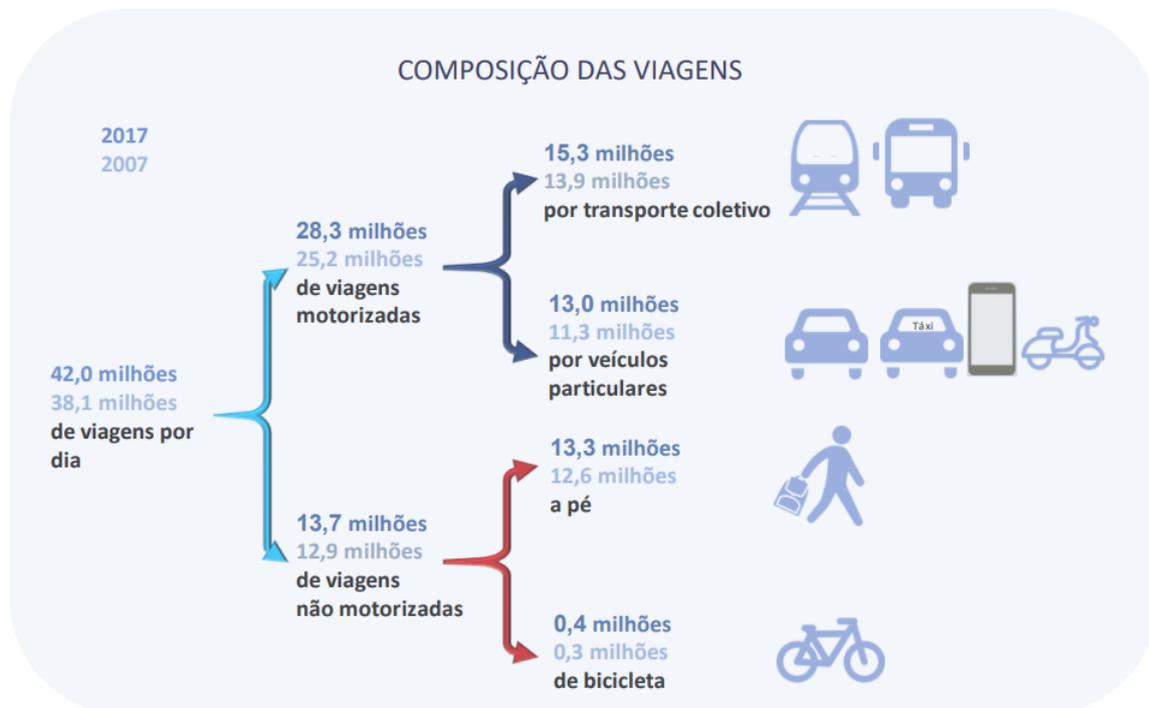


Figura 3.3: Composição das viagens segundo a Pesquisa OD 2017

Fonte: OD (2019)

3.1.4 Viagens multimodais

A Pesquisa OD 2017 avalia também viagens com múltiplos modais envolvidos. Para isso, a base de dados possui alguns campos que permitem registrar até quatro modais. A Tabela 3.2 mostra esses campos.

Campo	Descrição
ZONA_O	Zona de Origem
ZONA_T1	Zona da 1ª Transferência
ZONA_T2	Zona da 2ª Transferência
ZONA_T3	Zona da 3ª Transferência
ZONA_D	Zona de Destino
CO_O_X	Coordenada X Origem
CO_O_Y	Coordenada Y Origem
CO_T1_X	Coordenada X 1ª Transferência
CO_T1_Y	Coordenada Y 1ª Transferência
CO_T2_X	Coordenada X 2ª Transferência
CO_T2_Y	Coordenada Y 2ª Transferência
CO_T3_X	Coordenada X 3ª Transferência
CO_T3_Y	Coordenada Y 3ª Transferência
CO_D_X	Coordenada X Destino
CO_D_Y	Coordenada Y Destino
MODO1	Modo 1
MODO2	Modo 2
MODO3	Modo 3
MODO4	Modo 4

Tabela 3.2: Campos de localização multimodais da Pesquisa OD 2017

Fonte: OD (2017a)

As variáveis ZONA_O e ZONA_D se referem às zonas de origem e destino de uma viagem, respectivamente. As variáveis ZONA_TX, com $X \in \{1, 2, 3, 4\}$ se referem às zonas de transferência entre modais, caso existam. Analogamente, os campos de coordenadas são atribuídos para origem, destino e transferências. As variáveis MODOX, com $X \in \{1, 2, 3, 4\}$ indicam qual modal foi utilizado em cada trecho da viagem, utilizando os códigos de modais indicados na Tabela 3.1.

Com essas variáveis é possível extrair apenas os trechos de viagens que nos interessam, com o modal bicicleta, para os casos de viagens multimodais.

3.1.5 Extração e tratamento dos dados das viagens

Para extração dos dados referentes exclusivamente a viagens de bicicleta é necessário separar viagens com um único modal, que seja a bicicleta; e viagens multimodais, no qual um dos modais é a bicicleta.

Para viagens multimodais extraiu-se apenas os trechos referentes ao percurso percorrido de bicicleta, obtendo-se as novas zonas de origem, destino e as duas coordenadas

correspondentes.

Vale lembrar também o campo de fator de expansão da viagem, FE_VIA , que deve ser levado em consideração para que cada viagem da amostra seja convertida em um número de viagens representativas da população.

Foi também necessário converter o sistema de coordenadas, que na base de dados da Pesquisa OD 2017 utilizam o sistema $EPSG:22523$, específico para a região brasileira, para o sistema global $EPSG:4326$, que utiliza as conhecidas latitude e longitude, devido ao seu uso ser necessário nas bibliotecas de manipulação de dados geoespaciais.

Com isso, os dados estão prontos para serem manipulados pela ferramenta BikeScienceWeb.

3.2 Áreas de segregação

No modelo adotado, nós agregamos conjuntos de viagens com origem e destino numa mesma área. Isso é feito para facilitar a visualização com fluxos que representam muitas viagens, caso contrário, o elevado número de viagens torna a visualização e análise difícil. Além disso, isso permite também analisar os dados sumarizados nas áreas de segregação.

Para isso é necessário atribuir para cada viagem uma área de origem e uma de destino, logo, devemos particionar a área a ser analisada. Foram utilizados dois tipos de particionamento do mapa: (1) Zonas OD 2017 e (2) grade (ou *grid*).

As Zonas OD correspondem ao particionamento utilizados pela Pesquisa OD 2017, mostrado anteriormente na Figura 3.2. A Figura 3.4 mostra essas mesmas zonas diretamente na ferramenta BikeScienceWeb.

A outra opção, grade, consiste em particionar uma área bidimensional retangular em $N \times N$ células retangulares de mesmo tamanho. A área inicial é definida com parâmetros de deslocamento em cada direção em relação a um valor padrão que cobre a região metropolitana de São Paulo, que podem ser vistos na Figura 3.5. Um parâmetro adicional de tamanho define a quantidade total de células retangulares da partição.

A análise por meio das zonas é interessante pois sua escolha pelo Metrô de São Paulo leva em conta fatores de similaridade das condições urbanas na região. Porém, sua análise impede a visualização de viagens com origem e destino dentro de uma mesma zona. Para isso, a divisão do mapa em uma grade pode ser uma alternativa às Zonas OD, permitindo análises com diferentes níveis de detalhamento em uma mesma área através da variação da quantidade de células que compõe a grade.

3.3 Fluxos

Os fluxos são utilizados como uma abstração para representar um conjunto de viagens da Pesquisa OD 2017. Eles são representados por arcos dirigidos saindo de uma área de origem e apontando para uma área de destino, como mostrado na Figura 3.6, permitindo,

assim, visualizar e comparar a quantidade de viagens que ocorre entre as áreas. Em outras palavras, o modelo usa um único fluxo para representar todas as viagens entre duas áreas distintas.

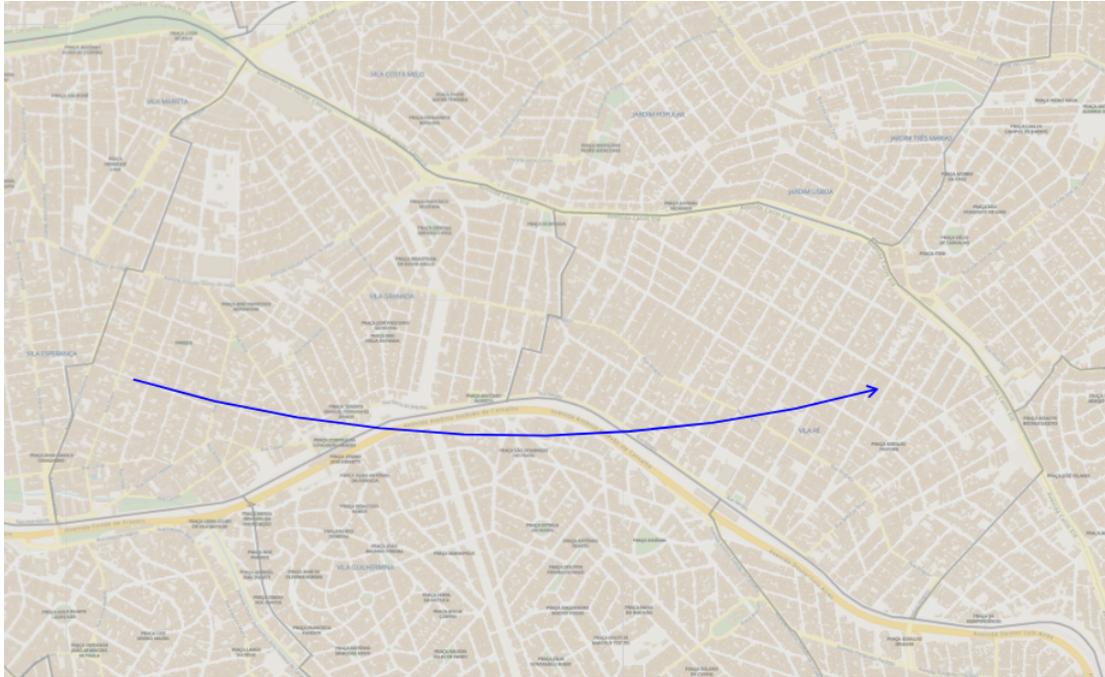


Figura 3.6: Um exemplo de fluxo no BikeScienceWeb

Para cada par de células (*area_de_origem*, *area_de_destino*) possível, são calculadas quantas viagens (*origem*, *destino*) foram realizadas com *origem* \in *area_de_origem* e *destino* \in *area_de_destino* e essa contagem resulta no total de viagens do fluxo (KON *et al.*, 2020). A espessura de um fluxo é proporcional à quantidade de viagens que ele representa, assim, fluxos mais carregados possuem uma espessura maior, enquanto fluxos com poucas viagens possuem espessura reduzida. O ponto de origem e de destino de um fluxo é o centro de massa dos pontos de origem e destino, respectivamente, das viagens que ele representa, ponderados pela fator de expansão da viagem.

Uma viagem é caracterizada pelas coordenadas geográficas de seus pontos de origem e destino. Logo, é necessário atribuir esses pontos às áreas às quais eles pertencem através de um processamento geográfico. Para isso, utilizamos bibliotecas de ciência de dados como Pandas para processamento numérico e o GeoPandas para processamento geográfico. Como consequência, os fluxos mostrados dependem da forma escolhida para dividir o mapa. Quanto menores as áreas, mais fluxos são gerados, o que permite análises mais detalhadas, mas que eventualmente podem dificultar a visualização no caso de haver muitos fluxos.

Por tratarmos as viagens como abstrações (fluxos), em nosso modelo nos interessam apenas as viagens que vão de uma área de origem para uma área de destino distintas, para permitir análises com diferentes níveis de detalhamento. Por isso, nossa análise descarta viagens circulares, com origem e destino na mesma área.

São descartadas também viagens com origem ou destino fora da área considerada na

grade, de modo a permitir visualizar apenas as viagens na região de interesse. Este caso não ocorre ao utilizar as divisões de mapa com Zonas OD, pois uma viagem sempre terá sua origem e destino dentro da área de interesse.

KON *et al.* (2020) destaca também que mostrar todos os fluxos resultantes para o usuário simultaneamente gera uma elevada quantidade de fluxos, o que dificulta uma análise razoável. Para mostrar a informação de forma mais compreensível, os dividimos em quartis, cada um contendo 25% do total de viagens. O primeiro quartil contém os fluxos mais intensos, ou seja, poucos fluxos mas cada um contendo muitas viagens, e assim por diante até o último quartil, que tem muitos fluxos, mas cada um com poucas viagens.

3.4 Filtros de viagens

As viagens que geram os fluxos podem ser filtradas na BikeScienceWeb, permitindo assim analisar segmentos da amostra baseadas em comportamento ou em informações demográficas.

Na Tabela 3.3 apresentamos os campos correspondente aos filtros, o nome da variável na base de dados e seus possíveis valores na Pesquisa OD 2017.

Variável	Conteúdo	Valores
<i>H_SAIDA</i>	Hora de Saída	Hora de Saída
<i>MIN_SAIDA</i>	Minuto de Saída	Minuto de Saída
<i>H_CHEG</i>	Hora de Chegada	Hora de Chegada
<i>MIN_CHEG</i>	Minuto de Chegada	Minuto de Chegada
<i>DURACAO</i>	Duração da Viagem	(em minutos)
<i>PE_BICI</i>	Por Que Viajou A Pé ou Bicicleta	1 - Pequena distância
		2 - Condução cara
		3 - Ponto/Estação distante
		4 - Condução demora para passar
		5 - Viagem demorada
		6 - Condução lotada
		7 - Atividade física
		8 - Outros motivos
<i>SEXO</i>	Sexo	1 - Masculino
		2 - Feminino
<i>IDADE</i>	Idade	(anos)
<i>RENDA_FA</i>	Renda Familiar Mensal	9 dígitos 2 casa decimais

Tabela 3.3: Filtros da Pesquisa OD utilizados na BikeScienceWeb

Fonte: OD (2017a)

Com isso, os filtros na BikeScienceWeb são selecionados na interface do usuário, são aplicados sobre as viagens no servidor para gerar os fluxos separados por quartis, e retornados à interface do usuário no mapa.

Os fluxos na ferramenta foram separados em categorias para efeito de organização e são apresentados na Tabela 3.4.

Categoria	Filtro	Opções
Filtros temporais	Horário de Início	Manhã (6h a 12h) Tarde (12h a 18h) Noite (18h a 23h) Horário específico
Filtros temporais	Horário de término	Manhã (6h a 12h) Tarde (12h a 18h) Noite (18h a 23h) Horário específico
Filtros de viagem	Duração	Intervalo entre 1 a 180 minutos Curta distância Preço alto do transporte público Difícil acesso ao transporte público
Filtros de viagem	Motivos da viagem	Lentidão do transporte público Viagem muito demorada Transporte público lotado Atividade física Outros
Filtros demográficos	Sexo	Homens Mulheres
Filtros demográficos	Renda familiar	até R\$ 1908 de R\$ 1908 até R\$ 3816 de R\$ 3816 até R\$ 7632 de R\$ 7632 até R\$ 11448 mais de R\$ 11488

Tabela 3.4: Filtros e suas categorias na BikeScienceWeb
Fonte: OD (2017a)

3.5 Camadas

As camadas auxiliam a visualizar informações adicionais que complementam a análise dos fluxos.

Os arquivos de camadas utilizados são do tipo *shapefile*, que em ESRI (2021) é definido como um formato que armazena geometrias não topológicas e informações de atributos para dados geoespaciais. O formato suporta pontos, linhas e áreas (representadas por polígonos fechados).

A Tabela 3.5 mostra todas as camadas utilizadas na BikeScienceWeb. Para a obtenção dos *shapefiles* citados, foram usadas como fontes a CET-SP (2021b) para a Rede Cicloviária e o CET-SP (2021a) para dos demais.

Categoria	Camada	Descrição
Transportes de alta capacidade	Linhas da CPTM	Linhas com os caminhos feitos pelas linhas da CPTM
Transportes de alta capacidade	Estações da CPTM	Pontos geográficos com as localizações das estações da CPTM
Transportes de alta capacidade	Linhas do Metrô	Linhas com os caminhos feitos pelas linhas de Metrô
Transportes de alta capacidade	Estações do Metrô	Pontos geográficos com as localizações das estações de Metrô
Infraestrutura cicloviária	Rede Cicloviária	Linhas indicando a localização da estrutura cicloviária (ciclovias, ciclorrotas, ciclofaixas)
Ocorrências de trânsito	Ocorrências com bicicletas	Pontos indicando os acidentes de trânsito envolvendo um ciclista.

Tabela 3.5: *Camadas utilizadas na BikeScienceWeb*

3.6 Cálculo dos Fluxos

Considere a Figura 3.7, que contém uma grade simplificada 2×2 . Nela são representadas duas viagens pelos vetores $\overrightarrow{O_1D_1}$ e $\overrightarrow{O_2D_2}$. As origens das duas viagens pertencem à célula $ABCD$ da grade e os destinos pertencem à célula $CFGH$. Como elas possuem as mesmas áreas de origem e de destino, representaremos essas viagens por um fluxo OD, na Figura 3.7, o vetor $\overrightarrow{O_{od}D_{od}}$.

O fluxo OD terá origem O_{od} no centro de massa de O_1 e O_2 , e destino D_{od} no centro de massa de D_1 e D_2 , ponderados pelo fator de expansão da via (FE_VIA). Ou seja,

$$O_{od} = \frac{O_1 \times FE_VIA_1 + O_2 \times FE_VIA_2}{FE_VIA_1 + FE_VIA_2}$$

$$D_{od} = \frac{D_1 \times FE_VIA_1 + D_2 \times FE_VIA_2}{FE_VIA_1 + FE_VIA_2}$$

Vamos supor, por simplicidade, que os fatores de expansão das duas viagens na Figura 3.7 sejam iguais. Então o fluxo OD terá origem no ponto médio de O_1 e O_2 e destino no ponto médio de D_1 e D_2 , como representado na figura. O fator de expansão resultante é a soma das contribuições de cada viagem, para cada par de células distintas da grade.

Vamos formalizar o algoritmo para o caso geral. Os algoritmos são baseados em [Kon et al. \(2020\)](#), porém, modificados para não considerar estações de bicicleta.

Considere um conjunto de viagens \mathcal{T} e uma grade geográfica $\mathcal{G}_{M,N}$ de $M \times N$ células. O algoritmo calcula o conjunto de fluxos \mathcal{F} correspondente.

Cada viagem $t \in \mathcal{T}$ é uma tupla com as informações da Pesquisa OD17. Cada célula $c_{ij} \in \mathcal{G}$ é um polígono. Na saída do algoritmo, cada fluxo $f \in \mathcal{F}$ é uma tupla contendo as células de origem e destino do fluxo, as coordenadas de sua origem e destino e seu peso. O

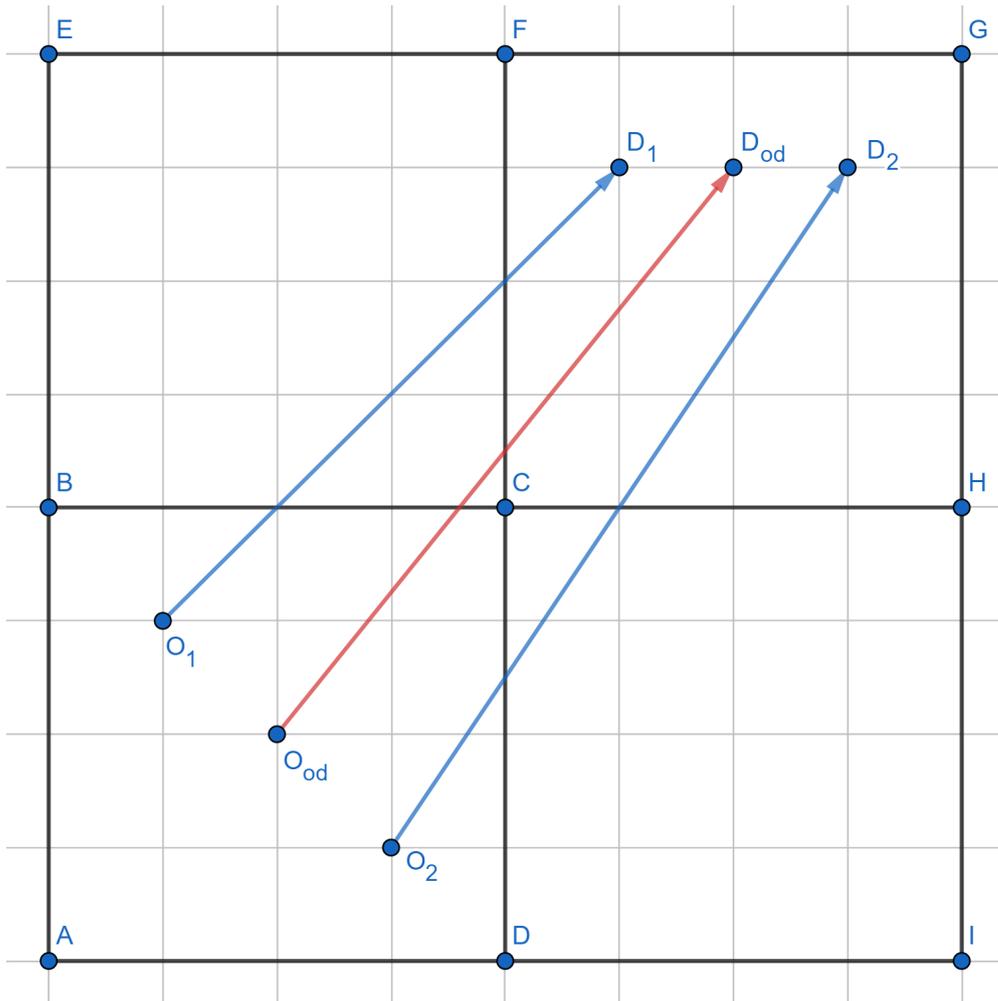


Figura 3.7: Um exemplo de cálculo de fluxo

peso de um fluxo é a soma das viagens (FE_VIA) que ele representa.

O Programa 3.1 recebe uma grade \mathcal{G} e um conjunto de viagens \mathcal{T} . Em seguida, ele realiza dois *spatial joins* (linhas 5 e 10) para determinar os índices das células da grade que contém a origem e o destino das viagens, respectivamente (i_start, j_start) e (i_end, j_end) . O algoritmo então retorna uma chamada à função *WeightedCounts* (linha 13) que calcula os fluxos, apresentada a seguir.

O Programa 3.2, recebe um conjunto de viagens, calcula o seu peso normalizado (linha 3), obtém as coordenadas das viagens com o peso normalizado (linhas 6-9) e, finalmente, obtém as coordenadas dos fluxos resultantes para cada par de células (linha 12) e retorna o resultado (linha 13).

O algoritmo para separar os fluxos em quartis é idêntico ao apresentado por [KON et al. \(2020\)](#).

Programa 3.1 Algoritmo 1: Cálculo dos fluxos.

```

1  FUNCTION AbstractFlows( $\mathcal{T}$ ,  $\mathcal{G}$ )
2    ▷ Encontra índices da célula que contém a origem da viagem
3     $trips \leftarrow \mathcal{T}$ 
4     $trips.geometry = trips.origin$  ▷ Define a geometria para o spatial join na origem
5     $grid\_trips = SpatialJoin(\mathcal{G}, trips)$ 
6     $grid\_trips.rename\_column(i, i\_start)$ 
7     $grid\_trips.rename\_column(j, j\_start)$ 
8    ▷ Encontra índices da célula que contém o destino da viagem
9     $grid\_trips.geometry = trips.destination$  ▷ Define a geometria para o spatial join no
     $destino$ 
10    $grid\_trips = SpatialJoin(\mathcal{G}, grid\_trips)$ 
11    $grid\_trips.rename\_column(i, i\_end)$ 
12    $grid\_trips.rename\_column(j, j\_end)$ 
13   return WeightedCounts( $grid\_trips$ )
14  end

```

Programa 3.2 Algoritmo 2: Cálculo dos centros de massa.

```

1  FUNCTION WeightedCounts( $grid\_trips$ )
2    ▷ Normaliza trip_counts com pesos entre 0 e 1
3     $centers \leftarrow grid\_trips.GroupBy(i\_start, j\_start, i\_end, j\_end).Normalize(column =$ 
     $trip\_count)$ 
4
5    ▷ Calcula as coordenadas das viagens com pesos normalizados
6     $centers.start\_lat \leftarrow centers.start\_lat \times centers.trip\_count$ 
7     $centers.start\_lon \leftarrow centers.start\_lon \times centers.trip\_count$ 
8     $centers.end\_lat \leftarrow centers.end\_lat \times centers.trip\_count$ 
9     $centers.end\_lon \leftarrow centers.end\_lon \times centers.trip\_count$ 
10
11   ▷ Calcula os totais para cada par (origem, destino) de células do grid
12    $centers \leftarrow centers.GroupBy(i\_start, j\_start, i\_end, j\_end).Sum(start\_lat, start\_lon,$ 
     $end\_lat, end\_lon)$ 
13   return  $centers$ 
14  end

```

3.7 Interface do usuário

A interface do usuário (UI) foi criada como um SPA utilizando o arcabouço Vue.js. Sua estrutura contém um menu lateral contendo abas para controlar as principais funcionalidades e o mapa, onde serão exibidos os fluxos e as camadas ativas, como pode ser visto na Figura 3.1.

A Figura 3.8 mostra a aba do Seletor Fluxos, nela podemos ver: (1) a opção de escolha das áreas de segregação (grade ou Zonas OD 2017); (2) o botão para alterar os parâmetros da grade; e (3) a seleção dos quartis de fluxos a serem exibidos no mapa.

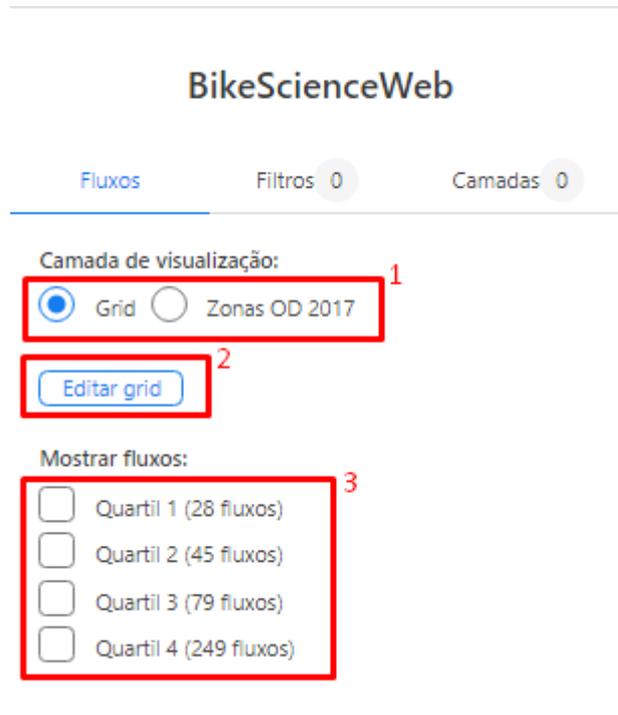


Figura 3.8: Aba do seletor de fluxos na UI



Figura 3.9: A edição da grade na UI

A Figura 3.9 mostra em (1) a edição dos parâmetros da grade (tamanho e deslocamentos) e em (2) é possível recalculer os fluxos após feitas as alterações. Assim que um parâmetro é alterado, o cliente faz uma chamada ao servidor que retorna as coordenadas da nova grade para ser renderizada no mapa.

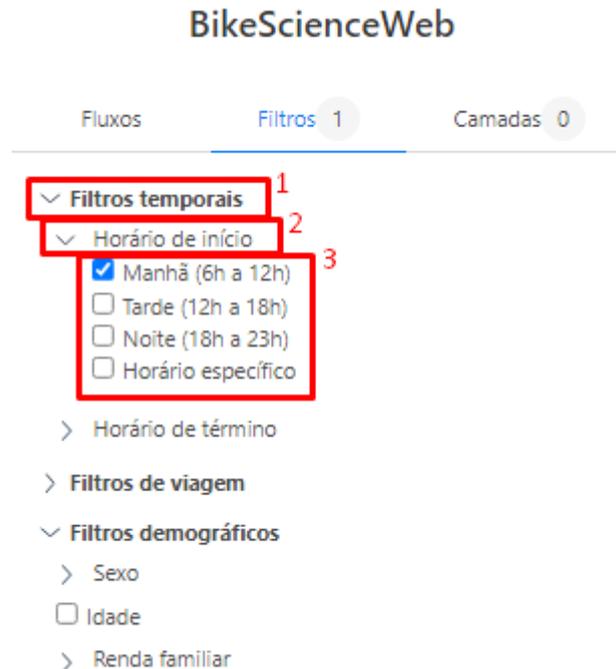


Figura 3.10: *Aba do seletor de filtros na UI*

A Figura 3.10 mostra a aba do Seletor de Filtros. Em (1) podemos ver as categorias de filtros, em (2) os filtros, e em (3) as suas opções de parâmetros. Basta clicar em uma categoria ou filtro para que as opções sejam renderizadas. Ao selecionar um parâmetro seu filtro é ativado automaticamente, a UI então faz uma chamadas à API que retorna os novos fluxos para serem renderizado no mapa.

Assim que a página é carregada, a UI faz uma chamada à API para receber todas as camadas de forma assíncrona para não afetar a experiência do usuário. A Figura 3.11 mostra a aba do Seletor de Camadas. Em (1) vemos as categorias de camadas e em (2) as camadas em si. Ao selecionar uma camada, a UI apenas altera o estado interno para mostrar a camada já carregada. O gerenciamento de estados na UI será detalhado na Seção 3.8.6.

3.8 Tecnologias utilizadas

Nesta seção, apresentamos as tecnologias envolvidas no desenvolvimento da BikeScienceWeb, descrevendo-as e citando como elas são utilizadas na ferramenta.



Figura 3.11: Aba do seletor de camadas na UI

3.8.1 Python

Python é uma linguagem de programação de código aberto de alto nível, interpretada, interativa e orientada a objetos. Ela incorpora módulos, exceções, tipagem dinâmica, tipos de dados de alto nível e classes (PYTHON, 2021).

Na BikeScienceWeb, a linguagem Python é, em conjunto com o arcabouço Flask, a base do *back-end*.

3.8.2 Flask

O Flask é um micro arcabouço para aplicações web projetado para desenvolver aplicações de forma simples e rápida, mas ainda com a possibilidade de escalar aplicações complexas. Para isso, o Flask suporta diversas extensões que permitem adicionar funcionalidades a uma aplicação como validação de formulários, manuseio de arquivos, tecnologias de autenticação, entre outros (FLASK, 2021).

No BikeScienceWeb o Flask é a base para a implementação do *back-end*.

3.8.3 Pandas

Pandas é uma biblioteca de código aberto para análise e manipulação de dados em Python, fornecendo estruturas de dados rápidas e flexíveis projetadas para tornar a manipulação de dados relacionais fácil e intuitiva (PANDAS, 2021). É considerada uma das ferramentas mais importantes de Python para cientistas de dados.

Na BikeScience é usada para obter os dados de arquivos no formato Valores separados por vírgulas (CSV) e para sua manipulação. Os dados de viagens são manipulados em pandas para aplicação dos filtros, obtenção do centro de massa dos fluxos, separação em quartis, entre outros.

3.8.4 GeoPandas

O GeoPandas é uma biblioteca de código aberto que estende os tipos de dados usados no Pandas para permitir operações espaciais em tipos geométricos. Suas operações geométricas são realizadas com o *shapely*, uma biblioteca para manipulação e análise de objetos geométricos planos. O GeoPandas depende também do Fiona para acesso de leitura e escrita a arquivos com dados geográficos e do matplotlib, uma para criação de gráficos interativos (GEOPANDAS, 2021).

Na BikeScienceWeb, o GeoPandas tem a função de obter os dados geográficos a partir dos *shapefiles* e, no caso das viagens, a partir do *DataFrame* obtido pela biblioteca Pandas. O GeoPandas também é utilizado para realizar os *spacial joins*, fundamentais para atribuir as áreas de origem e destino correspondentes a uma viagem.

3.8.5 Vue.js

Vue é um arcabouço de código aberto para construção de interfaces de usuário na Web. Ao contrário de outros arcabouços monolíticos o Vue foi projetado desde o início para ser adotado de forma incremental. Apesar de seu foco ser apenas na camada de visualização, possui fácil integração com outras bibliotecas ou projetos. Em combinação com bibliotecas de suporte o Vue é perfeitamente capaz de criar aplicativos de página única (VUEJS, 2021).

No BikeScienceWeb o Vue é a base para toda a implementação do *front-end*, integrando as implementações de HTML, CSS e JavaScript. Na implementação do projeto com aplicativo de página única utilizamos o suporte de algumas bibliotecas que serão discutidas adiante.

3.8.6 Vuex

Vuex é uma biblioteca de código aberto para gerenciamento de estado em aplicações Vue.js usando o padrão de projeto Observador. Em uma aplicação Vue tradicional, o gerenciamento de estado segue uma hierarquia onde o estado pode ser passado de uma componente para suas componentes filhas. Essa estrutura vertical dificulta o desenvolvimento de aplicações de página única, onde é desejável termos um estado compartilhado para que uma componente possa reagir a mudanças realizadas em outras.

O Vuex serve como um armazém de dados (ou *store*) centralizado para todas as componentes em uma aplicação, com regras garantindo que o estado só possa ser alterado de forma previsível (VUEX, 2021).

A Figura 3.12 mostra a estrutura de uma aplicação genérica implementada com Vue.js e Vuex. As componentes Vue possuem seu estado interno, mas acessam um estado global

gerenciado de forma independente, sem o poder de alterar esse estado global diretamente.

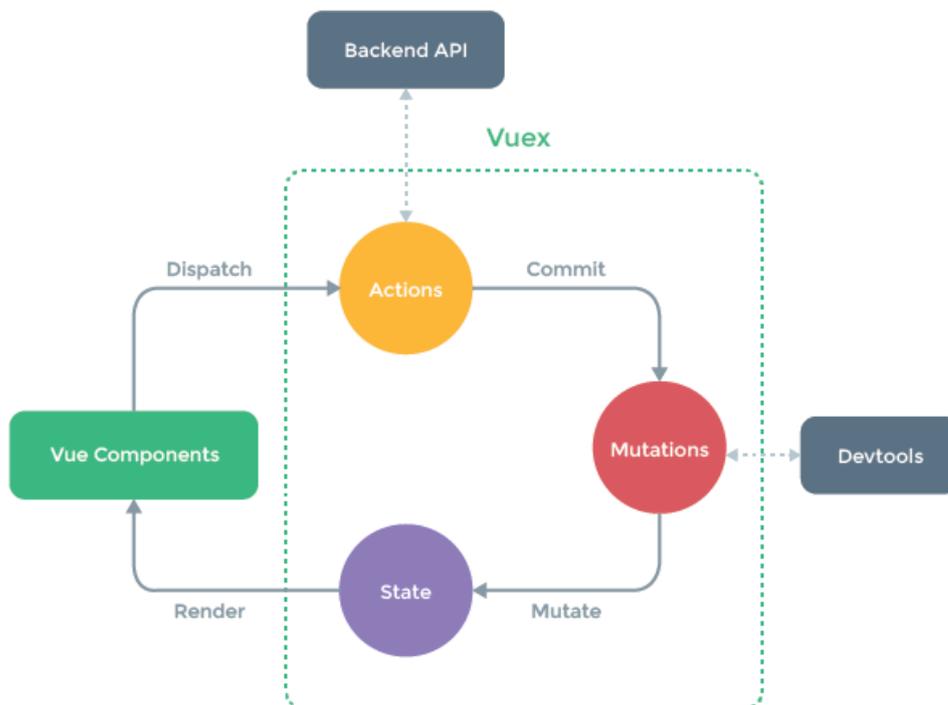


Figura 3.12: Estrutura genérica de um software com Vue.js e Vuex
 Fonte: VUEX (2021)

Uma componente pode emitir um sinal, ou *dispatch*, para que uma ação seja realizada. Uma ação pode conter operações arbitrárias, tanto síncronas quanto assíncronas, mas que nunca alteram o estado. Na ação faz-se então a confirmação, ou *commit*, de uma mutação.

Confirmar uma mutação é a única maneira de alterar o estado em uma *store* do Vuex. Após a alteração do estado pela mutação, as componentes Vue observando o estado serão automaticamente atualizadas.

3.8.7 Leaflet e Vue2Leaflet

O Leaflet é uma biblioteca JavaScript de código aberto para criação de mapas interativos projetada para funcionar de maneira eficiente tanto em plataformas desktop quanto mobile (LEAFLET, 2021).

O Vue-leaflet fornece componentes Vue para simplificar a criação de componentes de mapas do Leaflet no Vue (VUE2LEAFLET, 2021).

No BikeScienceWeb, tanto o Leaflet quanto o Vue2Leaflet são de extrema importância, dado que o núcleo da aplicação é o mapa fornecido e a interatividade que eles permitem alcançar.

3.8.8 Docker e Docker Compose

O Docker é uma plataforma de código aberto de containerização para criação e uso de contêineres Linux. Com ele é possível lidar com contêineres como se fossem máquinas virtuais modulares bastante leves (DOCKER, 2021).

O Docker simplifica o desenvolvimento, permitindo que desenvolvedores trabalhem em ambientes padronizados usando contêineres locais para suas aplicações e serviços (DOCKER, 2021). Os ambientes padronizados são importantes para garantir que o projeto seja executado sempre nas mesmas condições pelos desenvolvedores, utilizando as mesmas versões de dependências de um projeto, independente do sistema operacional e versões dos softwares utilizados localmente.

No BikeScienceWeb foram utilizados três contêineres Docker, descritos a seguir:

1. **ui**: contêiner com o *front-end* da aplicação, contendo a interface do usuário;
2. **backend**: contêiner com o *back-end*, contendo o serviço da Interface de Programação de Aplicações (API);
3. **nginx**: contêiner responsável por gerenciar a rede interna da aplicação.

O Docker Compose é uma ferramenta para definir e executar aplicações Docker com vários contêineres. Dessa forma, com um único comando é possível criar e iniciar todos os serviços necessários para uma aplicação (COMPOSE, 2021a).

O Docker Compose cria uma única rede interna para a aplicação que contém cada um dos contêineres, permitindo que eles possam ser acessados entre si (COMPOSE, 2021b).

Na próxima seção discutiremos as características da rede interna do BikeScienceWeb.

3.8.9 NGINX

O NGINX é um software leve de servidor HTTP de código aberto. Ele foi utilizado para gerenciar o tráfego em produção direcionando os acessos para o *front-end* ou *back-end* dependendo do seu Localizador Uniforme de Recursos (URL) acessado.

A Figura 3.13 mostra o funcionamento da rede interna do BikeScienceWeb, e como a rede fornecida pelo Docker Compose, em conjunto com o NGINX gerenciam o tráfego. Ao acessar a URL da aplicação na porta 80, porta padrão do protocolo HTTP, na máquina em produção, ocorre o redirecionamento do NGINX para a porta 9000 do container Docker do *back-end* caso a URL contenha o caminho */api*. As demais URL são redirecionadas da porta 80 da máquina de produção para a porta 80 do contêiner Docker do *front-end*. Dessa forma é possível ter os dois contêineres, *front-end* e *back-end*, funcionando em uma única máquina.

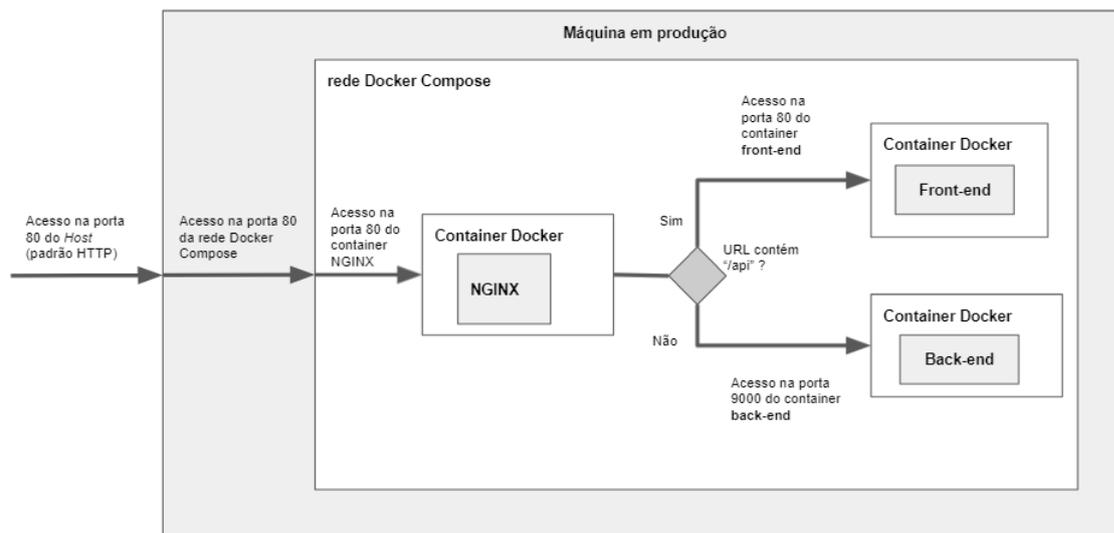


Figura 3.13: Rede interna do BikeScienceWeb

3.8.10 HTML, CSS e JavaScript

HTML (Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da web, utilizado para definir a estrutura e o significado do conteúdo da web. Geralmente é usado em conjunto com o CSS (Folhas de Estilo em Cascata) para descrever a aparência e o JavaScript para a funcionalidade de uma página web (MOZILLA, 2021).

Na BikeScience são utilizados HTML, CSS e JavaScript em conjunto com o *framework* Vue.js.

3.8.11 JSON

JSON (JavaScript Object Notation) é uma formatação leve para troca de dados baseada em um subconjunto da linguagem de programação JavaScript, é em formato texto e completamente independente da linguagem, pois usa convenções que são familiares a muitas linguagens. Essas propriedades fazem com que o JSON seja um formato ideal de troca de dados (JSON, 2021).

Na BikeScienceWeb o JSON é bastante utilizado como formato de troca de dados entre o *front-end* e o *back-end*.

3.8.12 NPM

NPM (Node Package Manager) é o gerenciador de pacotes do Node.js, criado em 2009 como um projeto código-aberto para auxiliar desenvolvedores JavaScript a compartilhar pacotes de códigos mais facilmente (NPM, 2021).

O NPM é o gerenciador de pacotes JavaScript utilizado na BikeScienceWeb.

3.8.13 Buefy

O Buefy é uma biblioteca de código aberto para componentes de interface do usuário feita com base no Vue.js e no Bulma, dois arcabouços bastante conhecidos na comunidade (BUEFY, 2021). A biblioteca permite a criação de componentes com estilos prontos que permitem agilizar o desenvolvimento de softwares.

No BikeScienceWeb o Buefy foi utilizado para criação das abas no menu principal, para os *inputs* de formulário com controle deslizante e para *inputs* de seleção de horário.

3.8.14 Toastr e Vue-toastr-2

O Toastr é uma biblioteca JavaScript de código aberto para notificações visuais que não interrompem o fluxo de navegação do usuário, sendo mostradas sem a necessidade de interação direta do usuário (TOASTR, 2021).

O Vue-toastr-2 também é uma biblioteca JavaScript de código aberto que simplifica o uso do Toastr com o Vue (VUE-TOASTR-2, 2021).

No BikeScienceWeb, o Toastr é utilizado para notificar ao usuário quando uma seleção de filtros é restritiva demais ao ponto de não gerar nenhum fluxo.

3.8.15 ESLint

O ESLint é uma biblioteca JavaScript de código aberto que analisa estaticamente o código para encontrar problemas rapidamente. Com ele também é possível definir regras de estilo para que os códigos fiquem padronizados entre vários desenvolvedores (ESLINT, 2021).

No BikeScienceWeb o ESLint foi utilizado no *front-end* para padronizar o estilo do código e agilizar o desenvolvimento ao lidar com erros de sintaxe.

3.9 Arquitetura da BikeScienceWeb

3.9.1 O modelo C4

O modelo C4 é uma abordagem de diagramação de arquitetura de software criada para ser de fácil aprendizado e amigável para desenvolvedores (BROWN, 2021). Ele é baseado na decomposição estrutural de um sistema de forma a permitir diferentes níveis de detalhamento, facilitando assim sua aplicação a diferentes perfis, como um *designer*, um potencial cliente, ou um desenvolvedor. O modelo se baseia em abstrações para refletir como os arquitetos de software e desenvolvedores pensam e constroem softwares.

O modelo fornece uma maneira para que as equipes de desenvolvimento de software possam comunicar de forma eficiente e eficaz a arquitetura do software em diferentes níveis de detalhe, contando histórias diferentes para cada tipo de público (BROWN, 2021). Segundo VÁZQUEZ-INGELMO *et al.* (2020), o modelo C4 pode ser visto como um complemento para facilitar a compreensão de diagramas na Linguagem de Modelagem Unificada (UML) e do processo de desenvolvimento de software.

Para obter esses diferentes níveis de abstração, são utilizados quatro tipos de diagramas, definidos em BROWN (2019) e apresentados a seguir.

(1) **Diagrama de Contexto do Sistema:** O diagrama de contexto mostra o sistema como uma caixa cercada pelos seus usuários e outros sistemas com os quais ele interage. O importante aqui não é detalhar tecnologias, protocolos, etc. Mas sim mostrar em qual contexto se encaixa o sistema de software.

(2) **Diagrama de Contêineres:** Um contêiner é, essencialmente, uma unidade que pode ser separada das demais e que executa código ou armazena dados. O diagrama de contêineres mostra a arquitetura do software em alto nível e como as responsabilidades são distribuídas. Ele mostra também as principais escolhas de tecnologia e como os contêineres se comunicam entre si.

(3) **Diagrama de Componentes:** O diagrama de componentes mostra como um contêiner é composto por componentes, o que é cada uma dessas componentes, suas responsabilidades e a tecnologia utilizada.

(4) **Diagrama de Código:** Podemos mostrar o interior de uma componente usando diagramas de classe UML, diagramas de entidade-relacionamento ou similares. No modelo C4, este nível é considerado opcional, pois o criador considera que esse nível de detalhamento somente é recomendado para as componentes mais importantes. Mesmo nesses casos, ele recomenda apenas utilizar diagramas criados automaticamente por um Ambiente de Desenvolvimento Integrado (IDE) ou uma ferramenta de modelagem de UML.

Nas figuras 3.14 a 3.20 apresentamos os diagramas do modelo C4 para a BikeScienceWeb.

A Figura 3.14 mostra o diagrama de contexto para a BikeScienceWeb. Nela representamos o usuário interagindo diretamente com o sistema, o qual interage com o sistema externo de mapas, responsável por fornecer ao sistema as imagens do mapa em blocos, ou *tiles*.

Na Figura 3.15 apresentamos o diagrama de contêineres, expandindo o sistema nos contêineres *front-end* e *back-end*. O contêiner do *front-end* se comunica com o do *back-end* através de chamadas à API.

Na Figura 3.16 apresentamos o diagrama de componentes do contêiner *Front-end*. Nele mostramos como o usuário interage com cada componente. A componente, quando necessário, confirma uma ação para o gerenciador de estados (Vuex) que é responsável por alterar o estado do mapa, refletindo com as alterações realizadas pelo usuário.

A Figura 3.17 mostra os diferentes controladores presentes no contêiner *back-end* com as suas respectivas responsabilidades.

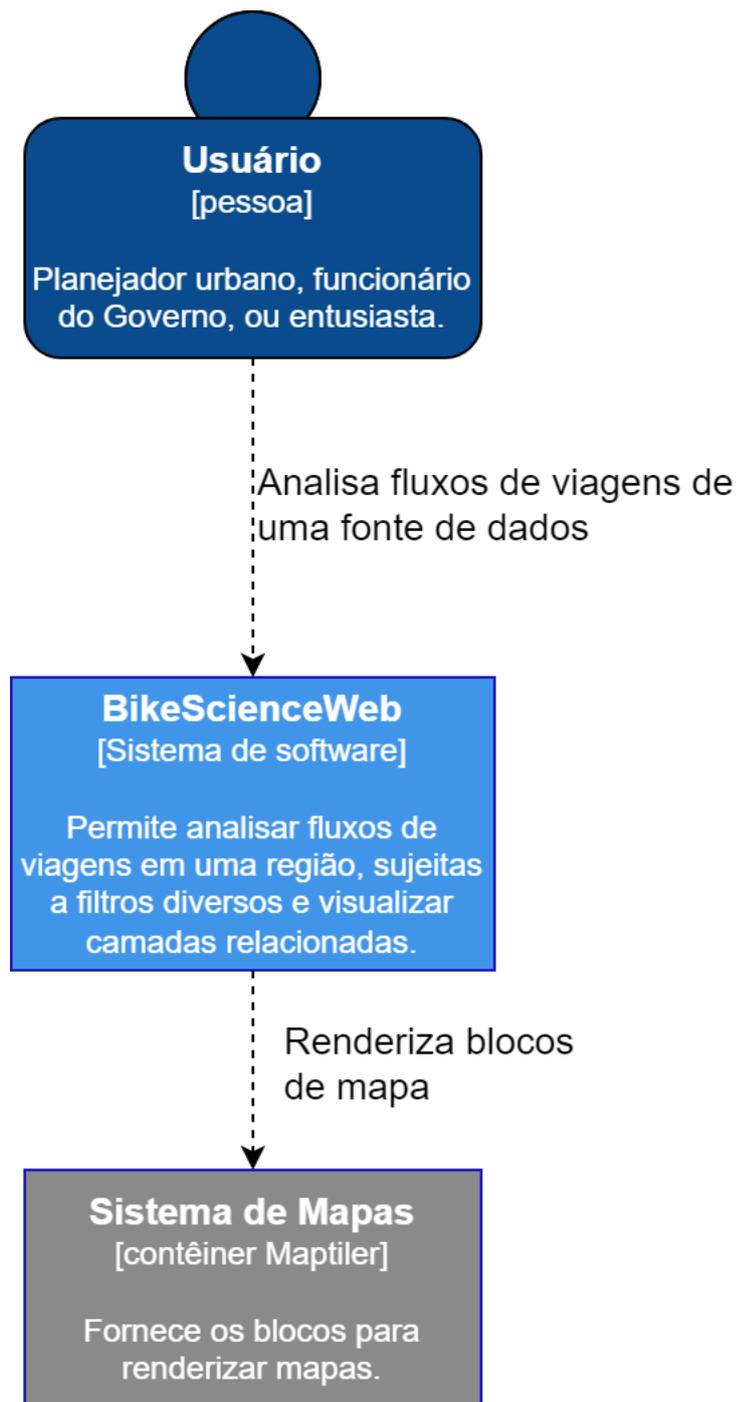


Figura 3.14: Diagrama de contexto para a BikeScienceWeb

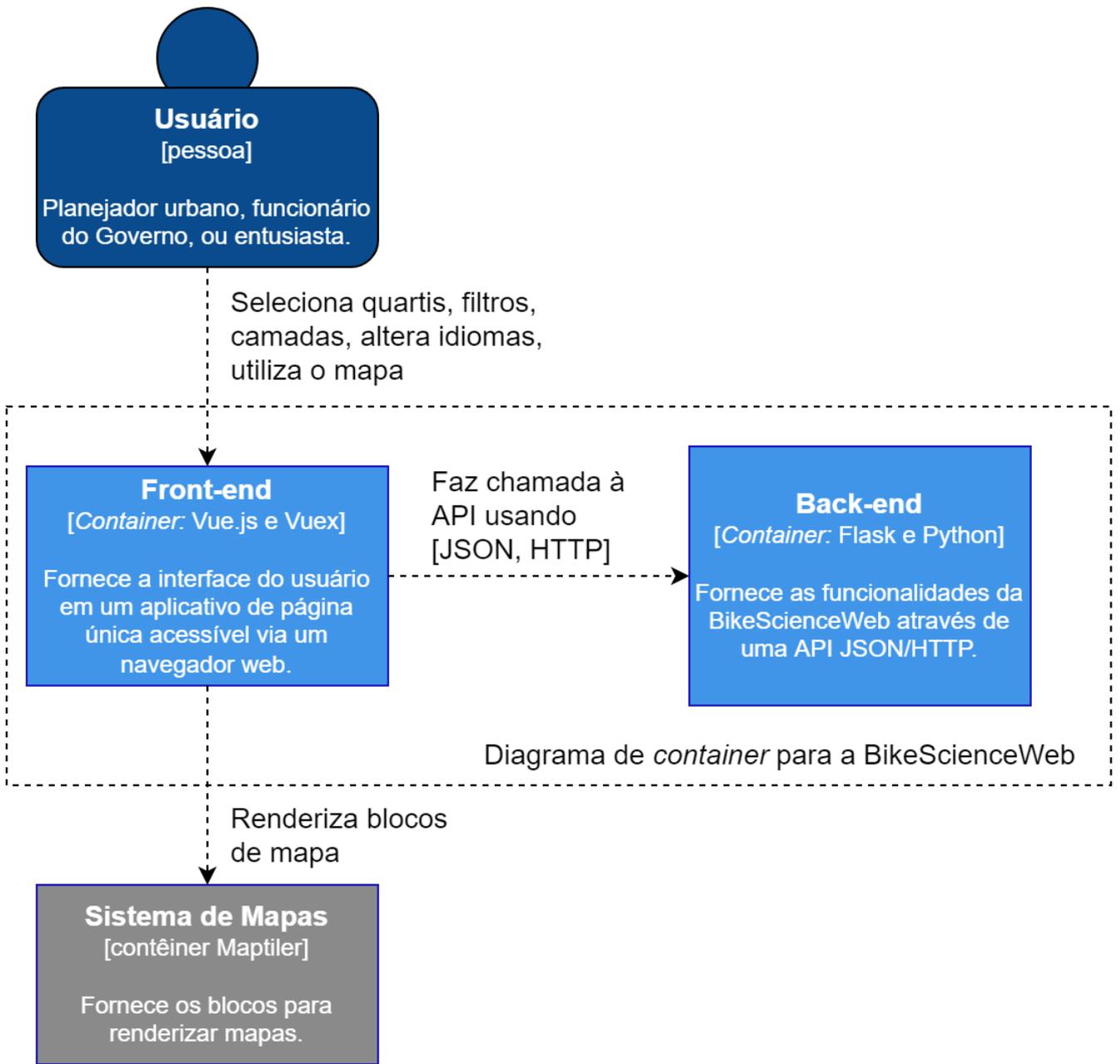


Figura 3.15: Diagrama de contêineres para a BikeScienceWeb

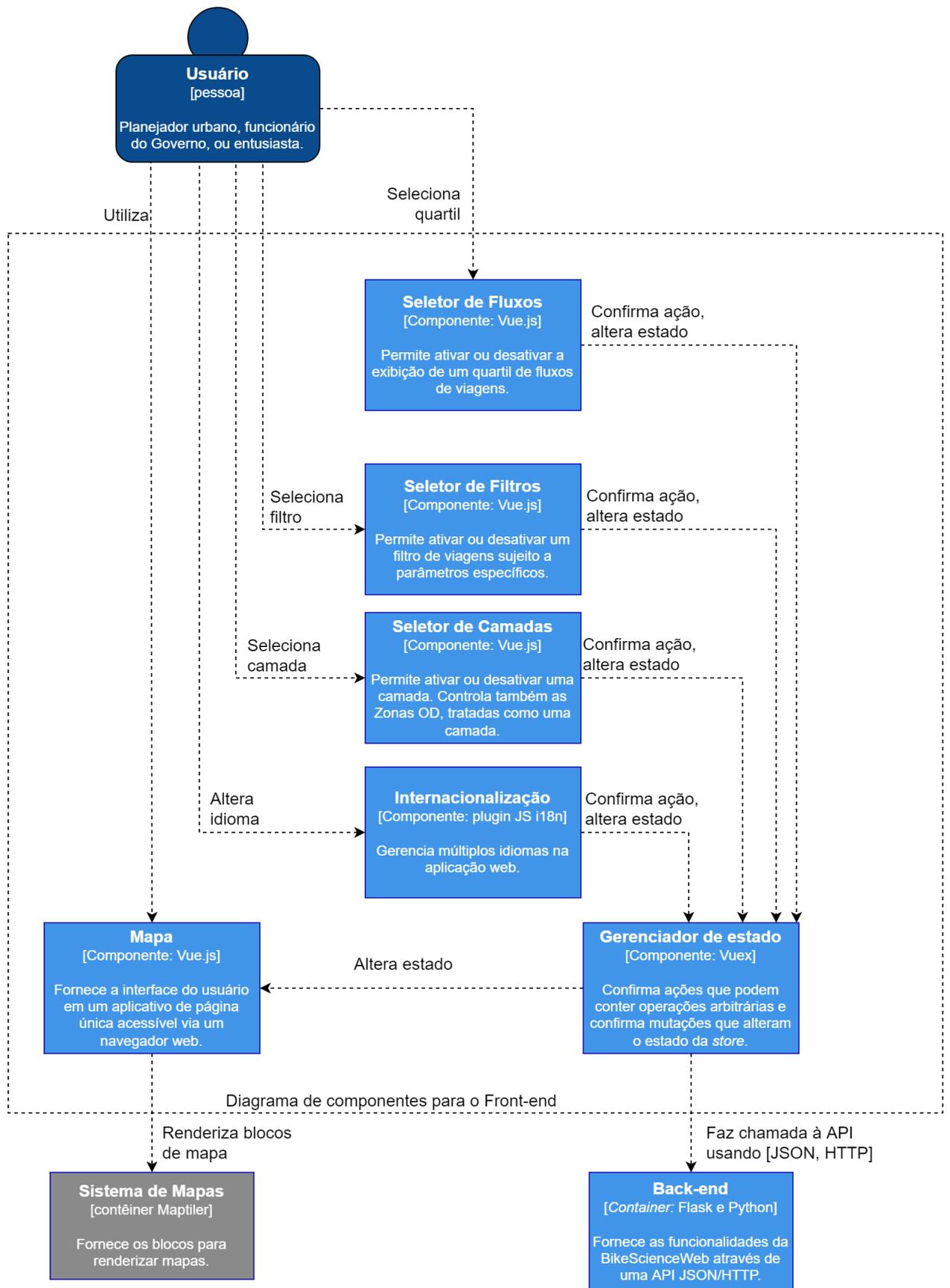


Figura 3.16: Diagrama de componentes do Front-end para a BikeScienceWeb

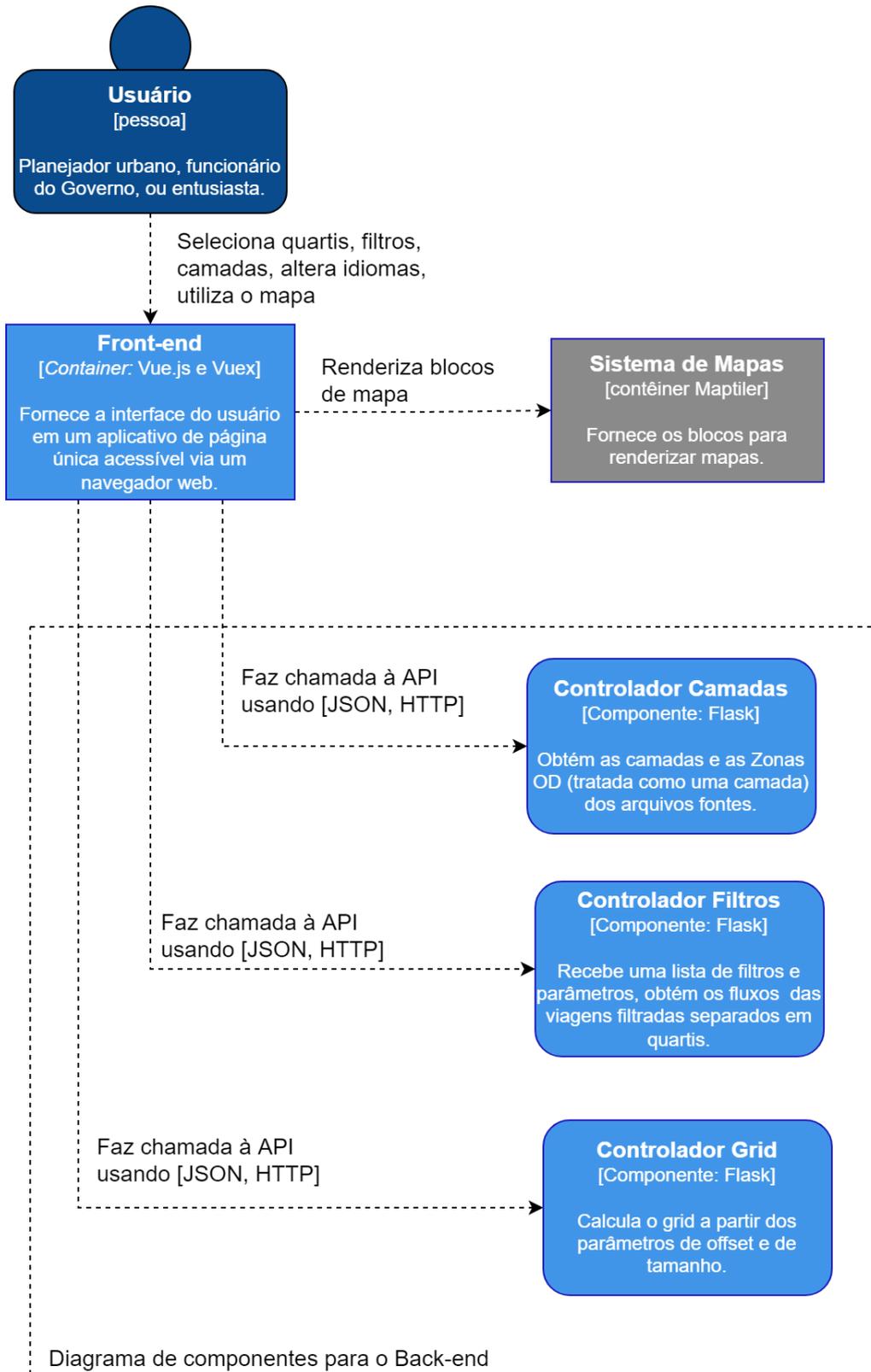


Figura 3.17: Diagrama de componentes do Back-end para a BikeScienceWeb

A Figura 3.18 mostra o diagrama de classes para a componente Controlador Filtros do contêiner *back-end*. A classe *ControladorFiltros* inicializa uma lista *Filtro[]* e a retorna para o *front-end*, que a utiliza para definir os parâmetros dos filtros escolhidos na interface do usuário. Ao ser instanciado, o *ControladorFiltros* ainda cria também um objeto *OD* contendo os dados da Pesquisa OD 2017. Quando o usuário requisita fluxos na interface do usuário, sejam eles filtrados ou não, é chamado o método *ControladorFiltros.handle_filtering()* via API, enviando o tipo de áreas de segregação (grade ou zonas) e os parâmetros dos filtros. A instância *OD* então instancia um objeto da classe *Grid* ou um da classe *Zones*. Ainda no método *handle_filtering()*, o *ControladorFiltros* executa cada um dos métodos para filtrar as viagens, desde que estejam ativos nos parâmetros. Aplicados os filtros, é chamado o método *coords_by_tiers()*, que aplica o método *grid_od()* ou *zones_od()*, dependendo do caso, que por sua vez chamam o método *abstractFlows()*, correspondente ao Programa 3.1, que realiza o cálculo dos fluxos com o auxílio do método *weightedCounts()*, que corresponde ao Programa 3.2. Em seguida, ainda no método *coords_by_tiers()*, é chamado o método *separate_in_tiers()* que separa os fluxos em quartis e os retorna.

A Figura 3.19 mostra o diagrama de classes para a componente Controlador Camadas do contêiner *back-end*. O *ControladorCamadas* basicamente instancia uma das classes de camadas, as quais possuem o método *load()*, que obtém os dados dos arquivos *shapefiles* específicos, e o método *get()* que chama o *load()* para retorná-lo.

A Figura 3.20 mostra o diagrama de classes para a componente Controlador Grid do contêiner *back-end*. O *ControladorGrid* recebe do *front-end* os parâmetros para criação da grade, instancia um objeto da classe *Grid* e retorna suas coordenadas.

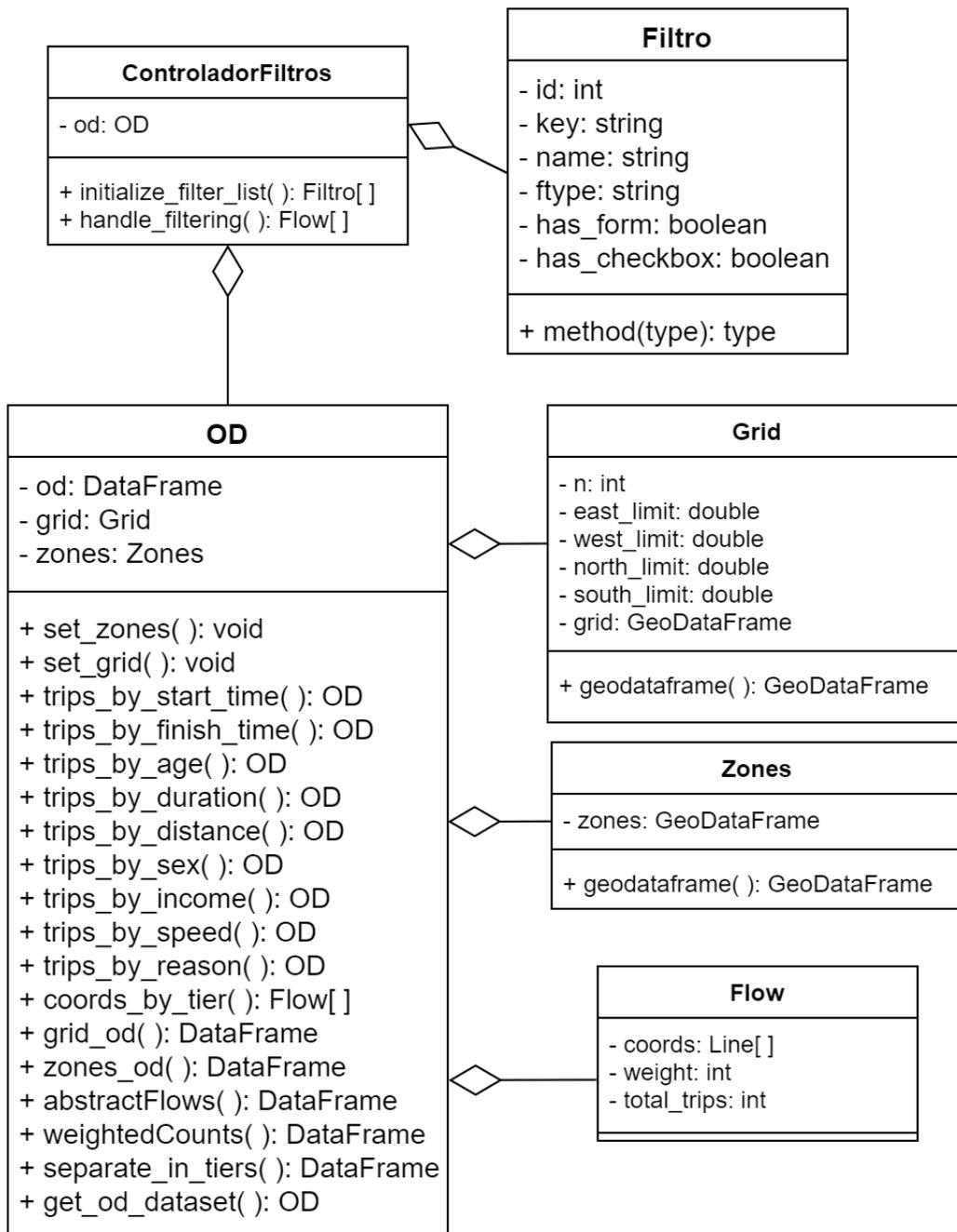


Figura 3.18: Diagrama do classes Controlador de Filtros

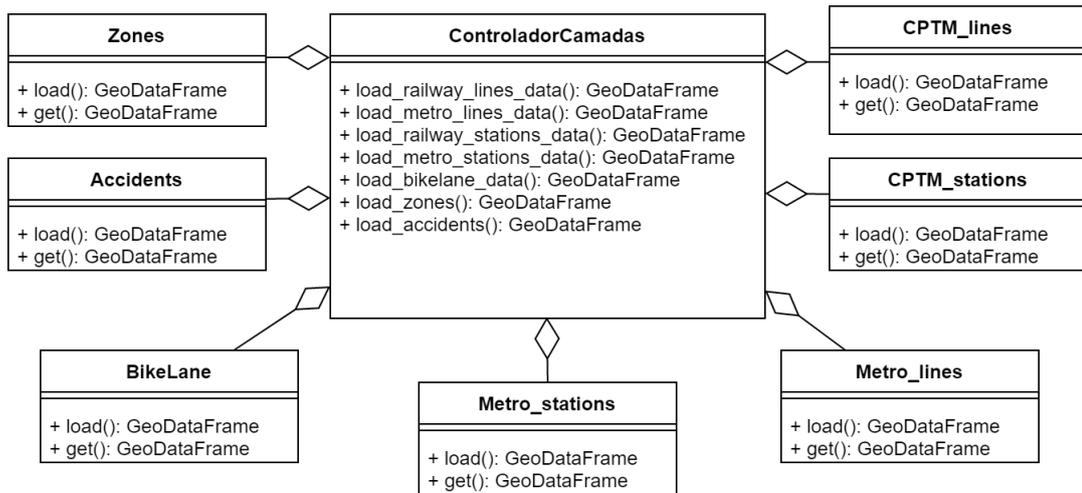


Figura 3.19: Diagrama de classes do Controlador de Camadas

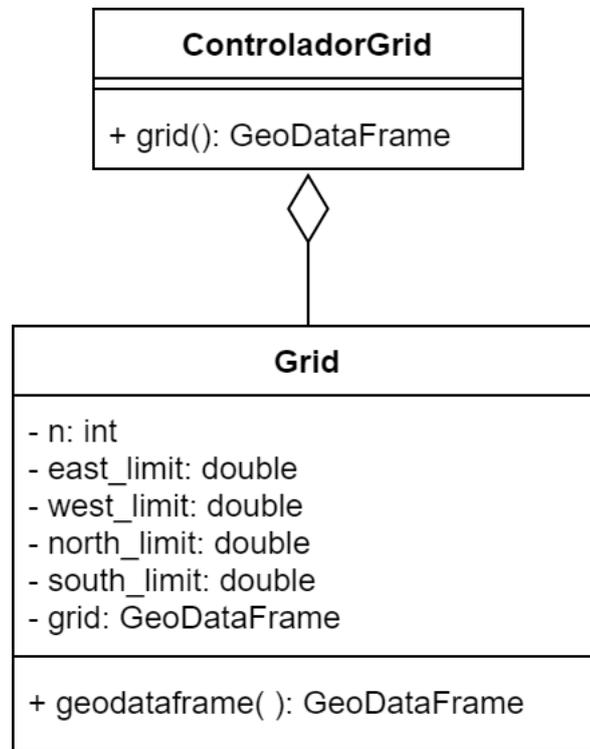


Figura 3.20: Diagrama de classes do Controlador de Grade

3.10 Ambiente de produção

Para subir o projeto em produção foi disponibilizada uma máquina virtual na Nuvem da Universidade de São Paulo (USP) com dois núcleos Intel(R) Xeon(R) CPU E7 - 2870 @ 2.40GHz de 64 bits, 4GB de memória *ram* utilizando o sistema operacional Ubuntu 20.04.

Os arquivos para seu funcionamento são importados diretamente via *Secure Shell* (SSH) e a implantação é feita com o uso de um *script* em *bash* que monta a imagem Docker de cada contêiner e interliga-os na rede interna do Docker Compose, gerenciada pelo NGINX. A configuração da rede interna da BikeScienceWeb é detalhada nas seções 3.8.8 e 3.8.9.

3.11 Exemplo de aplicação

Nesta seção vamos apresentar um exemplo de aplicação da ferramenta e de algumas análises que podem ser feitas com ela. Na Figura 3.21 selecionamos as Zonas OD como áreas de segregação, filtramos para mostrar apenas viagens com início durante a manhã (6h a 12h) e selecionamos todos os quartis de fluxos. Note a excessiva quantidade de fluxos, dificultando qualquer tipo de análise. Para visualizar os fluxos em uma grande área, a separação dos fluxos em quartis auxilia a verificar mais claramente quais deles são mais relevantes.

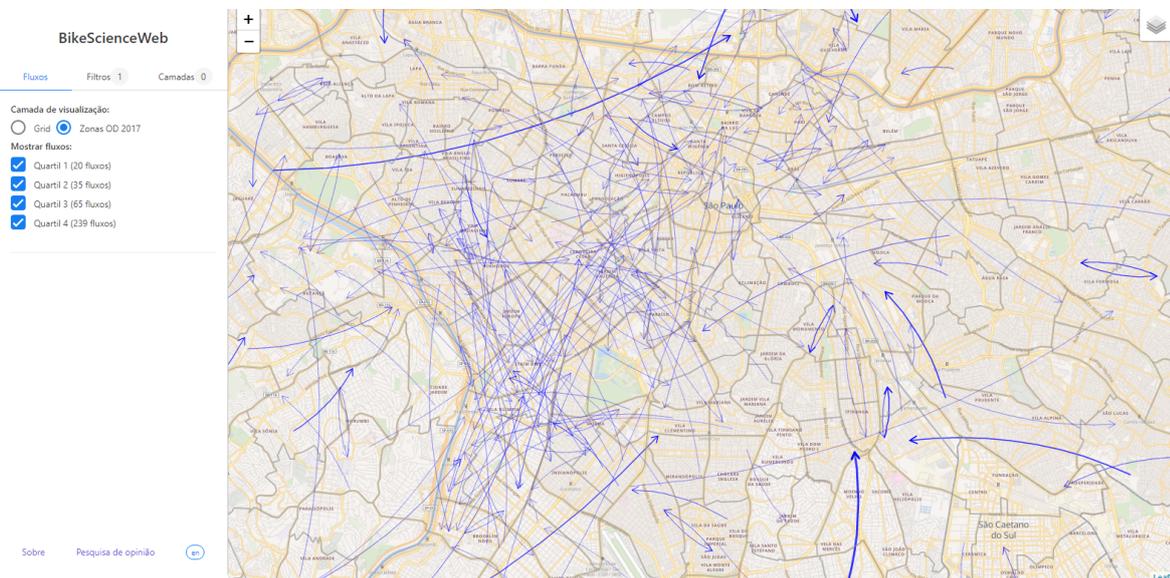


Figura 3.21: Quantidade excessiva de fluxos em São Paulo

A Figura 3.22 mostra a mesma situação, porém, focando no bairro Cerqueira César. Note que a quantidade de fluxos que passam pela região ainda é muito alta, tornando difícil saber se o fluxo apenas passa pela região ou se tem origem/destino ali.

Usar uma área por toda a cidade é útil para ter uma visão macro dos principais fluxos. Mas para analisar os fluxos dentro de uma região específica, o uso da grade como área

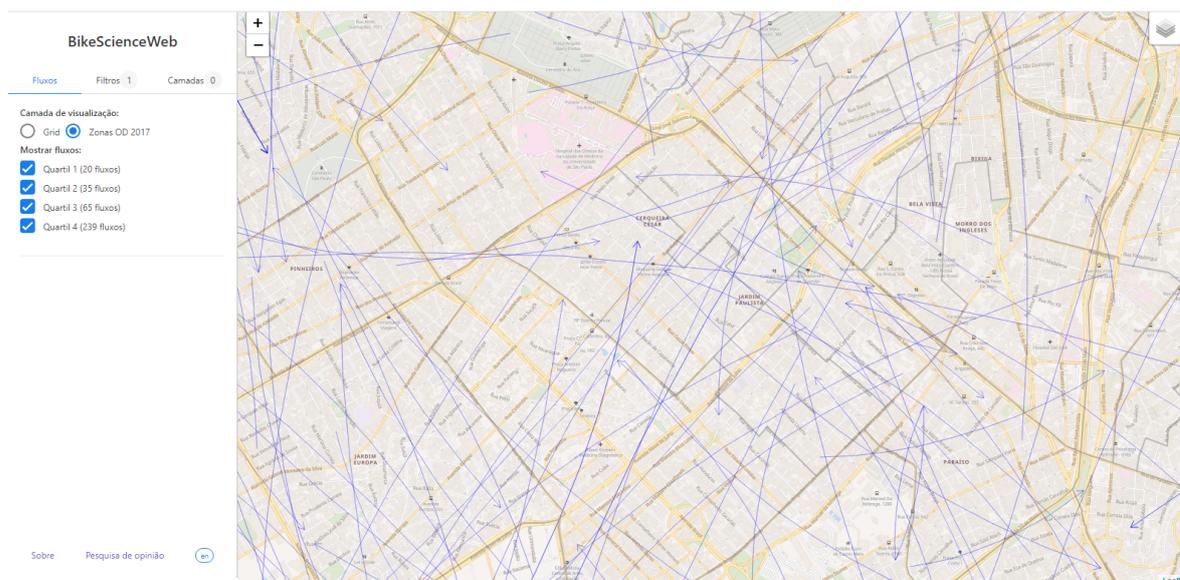


Figura 3.22: Quantidade excessiva de fluxos em Cerqueira César

de segregação pode fornecer uma visão mais clara. Na Figura 3.23 usamos as mesmas configurações do caso anterior, mas usando grade com limites que cercam o bairro. Note que a quantidade de fluxos é muito menor, mesmo com todos os quartis selecionados, permitindo ver facilmente os fluxos que começam e terminam no bairro.

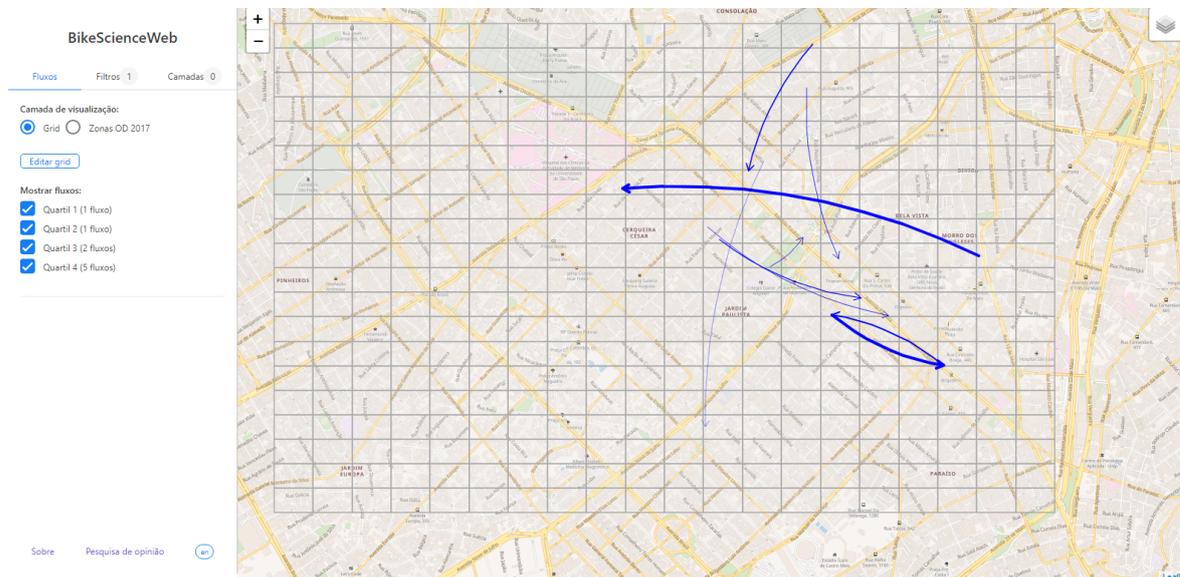


Figura 3.23: Fluxos em Cerqueira César

Na Figura 3.24, temos as mesmas configurações do caso anterior, porém adicionamos as camadas do Metrô e da infraestrutura cicloviária. Este caso é um ótimo exemplo de uma análise que pode influenciar na melhoria no ciclismo urbano. Identificar fluxos com difícil acesso a estações do Metrô, CPTM ou a uma ciclovia, por exemplo, pode direcionar o planejamento de melhorias na infraestrutura cicloviária priorizando os fluxos mais carregados. Outra análise possível é inserir os dados de ocorrência de acidentes com

3.11 | EXEMPLO DE APLICAÇÃO

bicicleta e propor alternativas mais seguras nessas regiões.

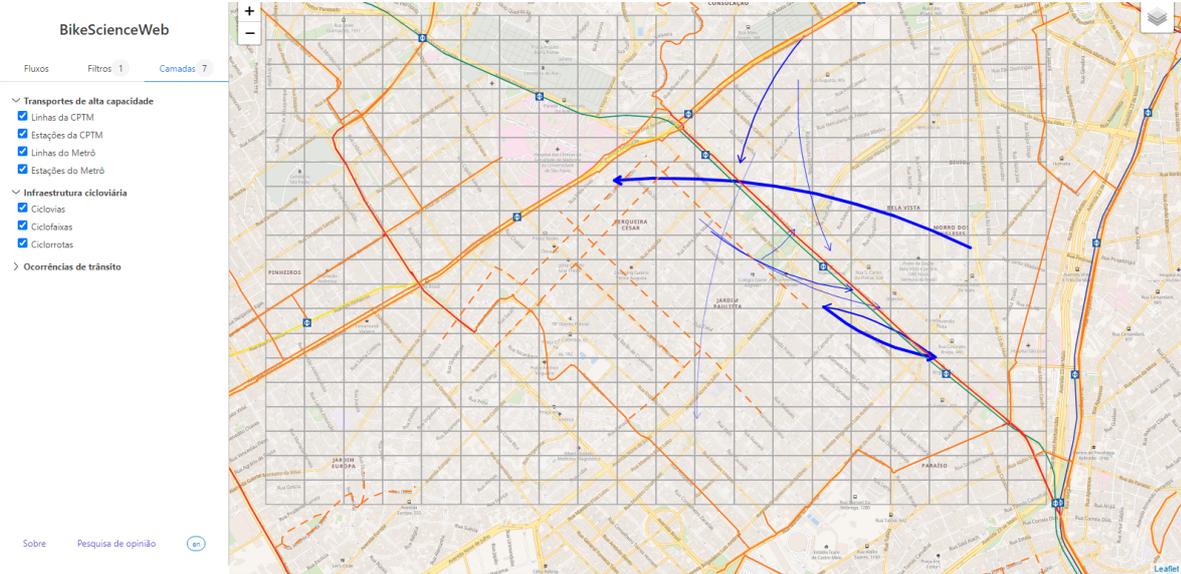


Figura 3.24: Fluxos em Cerqueira César com camadas de metrô e de infraestrutura ciclovitária

Capítulo 4

Avaliação

4.1 Metodologia

Com o objetivo de avaliar a aceitação da ferramenta BikeScienceWeb foi utilizado o *Technology Acceptance Model* (TAM) - Modelo de Aceitação Tecnológica. O TAM é um modelo bastante conhecido utilizado para avaliar como os usuários percebem a utilidade e a facilidade de uso de uma tecnologia (SOUZA, 2018). A utilidade percebida indica a opinião de uma pessoa sobre o quão útil é uma nova tecnologia para melhorar seu desempenho em suas atividades. A facilidade de uso percebida avalia o quão fácil é usar a nova tecnologia para realizar suas atividades (DAVIS, 1989).

O grupo escolhido para a realização da pesquisa inclui especialistas em mobilidade urbana, tais como ativistas que atuam junto a associações da sociedade civil ligadas à mobilidade urbana ou que atuam junto a organizações não governamentais e alguns funcionários do poder público da CET-SP já envolvidos com o projeto BikeScience. A pesquisa foi feita usando um questionário criado e aplicado via Google Forms baseado nas questões do TAM, entretanto, foi utilizada apenas uma questão de cada categoria do modelo ao invés das 10 questões originais. O envio do formulário aos participantes foi feito via *e-mail*. Entre os convidados que responderam a pesquisa tivemos 17 participantes.

Para as perguntas do TAM foi utilizada a escala Likert de 7 pontos, numerados de 1 (discordo totalmente) a 7 (concordo totalmente). Além disso, há perguntas abertas para que os participantes deixem sugestões de melhorias, ou novas funcionalidades, além de relatar problemas ou dificuldades encontradas ao utilizar a ferramenta. Finalmente, há também perguntas para identificar o perfil dos participantes.

4.2 Questionário

A Tabela 4.1 mostra as perguntas do questionário aplicado aos participantes separadas entre as categorias TAM: utilidade percebida (U1), facilidade percebida (F2) e intenção de uso (I3). A última categoria inclui as perguntas do perfil dos participantes (P4 a P9).

Os resultados da pesquisa são apresentados na Seção 4.3

Utilidade percebida	
U1	Acredito que o BikeScienceWeb melhoraria meu desempenho na análise desse tipo de dado em relação ao que eu consigo realizar hoje.
Facilidade de uso percebida	
F2	De modo geral, eu achei a BikeScienceWeb fácil de usar.
Intenção de uso	
I3	Se tiver acesso à BikeScienceWeb, eu pretendo usá-la para analisar dados sobre mobilidade urbana.
Sugestões do participante	
P4	Você tem sugestões de melhorias na ferramenta ou novas funcionalidades que gostaria que fossem nela implementadas?
P5	Descreva qualquer problema ou dificuldade que tenha tido ao usar a ferramenta.
Perfil do participante	
P6	Qual a sua área de atuação?
P7	Quantos anos de experiência você tem na área de atuação mencionada?
P8	Onde você trabalha ou atua?
P9	Qual seu gênero?
P10	Deixe seu <i>e-mail</i> caso deseje receber informação sobre os avanços da ferramenta. Essa informação não será divulgada.

Tabela 4.1: Perguntas do formulário de pesquisa de avaliação da BikeScienceWeb

4.3 Resultados

Os resultados das perguntas da avaliação TAM são apresentados na Figura 4.1. Em seguida, apresentamos os gráficos relativos ao perfil dos participantes. A Figura 4.2 apresenta as áreas de atuação dos participantes. A Figura 4.3 apresenta os anos de experiência dos participantes em sua área de atuação atual. A Figura 4.4 apresenta onde os participantes trabalham ou atuam com sua respectiva legenda na Tabela 4.2. Finalmente, a Figura 4.5 apresenta o gênero dos participantes. Nas perguntas P4 e P5 foram deixadas diversas sugestões de novas funcionalidades e de melhorias na usabilidade, além de dificuldades encontradas pelos participantes. Essas sugestões são apresentadas na Tabela 4.3. Os resultados são discutidos na Seção 4.4.

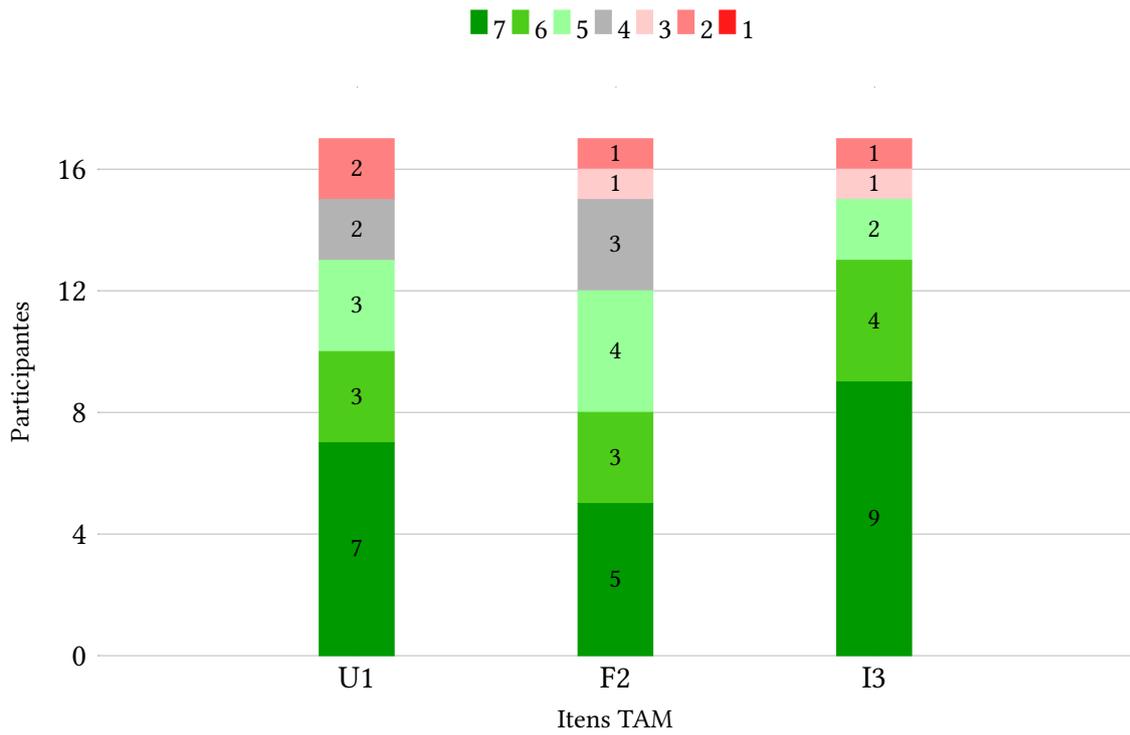


Figura 4.1: Respostas às perguntas da avaliação TAM.

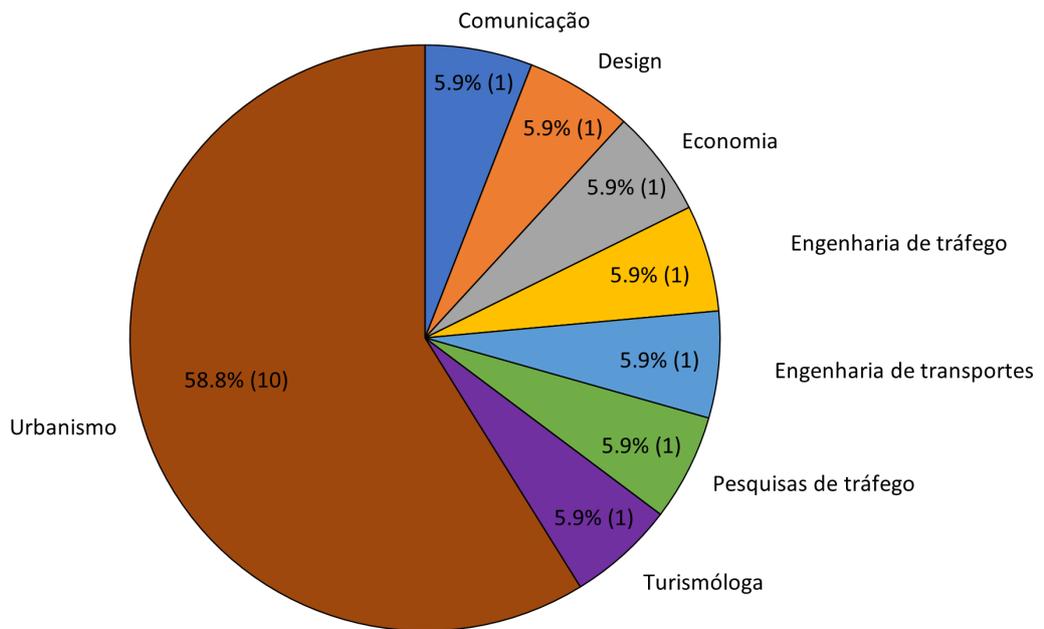


Figura 4.2: Áreas de atuação dos participantes.

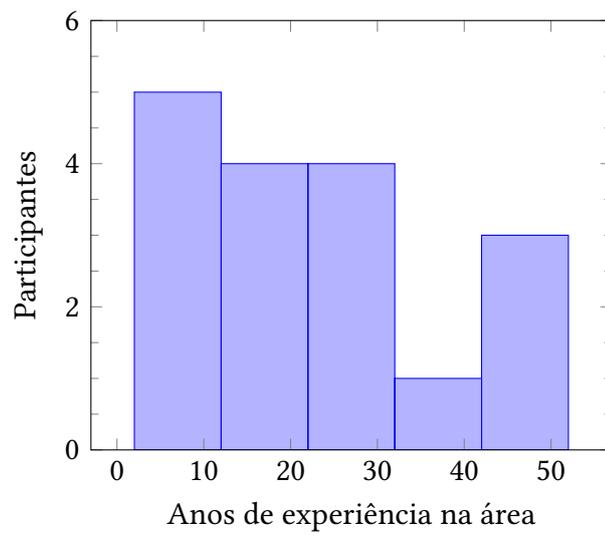


Figura 4.3: Anos de experiência dos participantes na área de atuação.

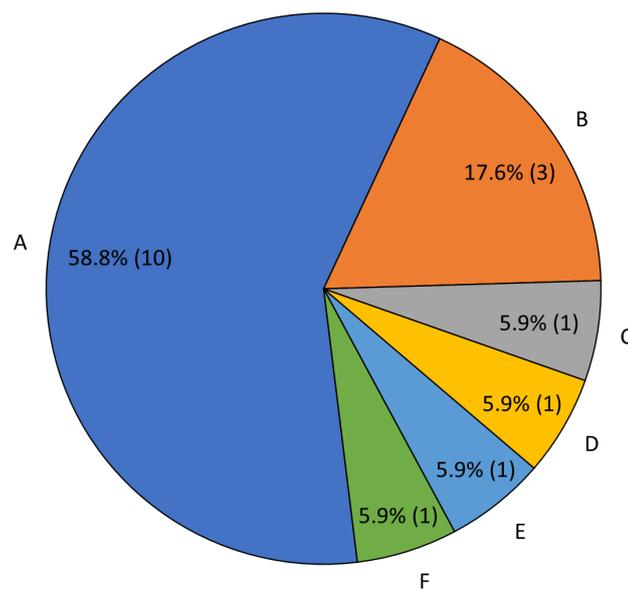


Figura 4.4: Trabalho ou atuação dos participantes.

Legenda	Descrição
A	Sou funcionário(a) do poder público
B	Sou ativista e/ou atuo junto a associações da sociedade civil ligadas à mobilidade urbana
C	Universidade e ativismo
D	Fui funcionária do poder público, atualmente sou ativista, atuo junto a organizações não governamentais e também atuo junto à Universidade em cursos de pós graduação
E	Atuo junto à Universidade (graduação, pós-graduação, professor ou pesquisador)
F	Sou ativista e atuo junto à startups de mobilidade.

Tabela 4.2: Trabalho ou atuação dos participantes

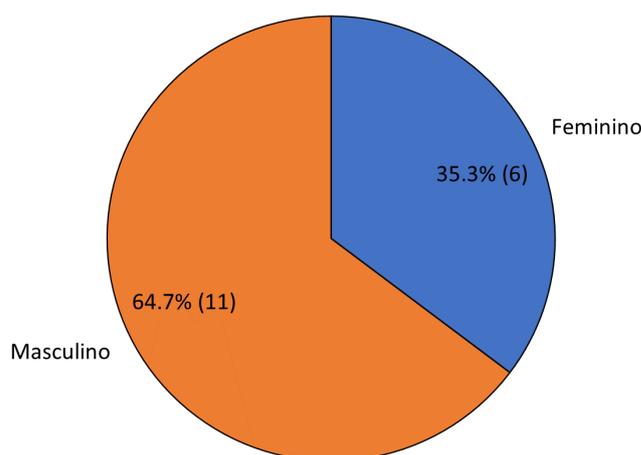


Figura 4.5: Gênero dos participantes.

4.4 Discussão dos resultados

Na Figura 4.1, verifica-se que para a utilidade percebida (U1) 13 dos 17 participantes avaliaram positivamente, enquanto que apenas 2 avaliaram negativamente. Para a facilidade de uso percebida (F2) 12 dos 17 participantes avaliaram positivamente e apenas 2 negativamente. Para a intenção de uso (I3) 15 dos 17 participantes avaliaram positivamente e apenas 2 negativamente.

A análise do perfil dos participantes mostra, na Figura 4.2, que a maioria deles atua na área de Urbanismo. As demais áreas tem apenas um representante, mas possuem relevância com o tema, como Pesquisa de tráfego, Engenharia de tráfego e Engenharia de transportes. A Figura 4.3 mostra que a experiência dos participantes é bastante variada, com alguns

Sugestões de novas funcionalidades
Poder comparar dois mapas lado a lado
Poder exportar os dados ou os mapas em CSV ou JSON para permitir utilizar o resultado da análise em outros contextos
Adicionar novos tipos de mapas como <i>spyder map</i> e mapa de calor
Incluir rotas de bicicletas
Incluir ocorrências de trânsito com pedestres
Identificar a data das camadas que são atualizadas mais frequentemente, como a infraestrutura cicloviária
Incluir a opção de usar os dados da Pesquisa OD 2007 para comparação
Incluir uma categoria de camadas referente à segurança, indicando incidência de furtos de bicicleta
Poder buscar um endereço no mapa
Descrever os limites na quantidade de viagens para cada quartil
Sugestões de usabilidade
Melhorar a visibilidade dos fluxos e dos limites das áreas de segregação
Poder alterar o volume ou a cor dos fluxos para melhorar a visualização
Tornar opcional a visualização das áreas de segregação
Poder optar por não recalcular os fluxos automaticamente ao selecionar um filtro
Mostrar quais quartis de fluxos estão ativos a todo momento
Dificuldades
Dificuldade em diferenciar os fluxos de diferentes quartis
Dificuldade em entender o fluxo e o que cada coisa significa
Dificuldade em encontrar os menus
Dificuldade em encontrar o guia do usuário
Lentidão nos carregamentos
Confusão devido ao cálculo automático dos fluxos ao alterar um filtro

Tabela 4.3: Respostas dos participantes às perguntas P4 e P5

casos de mais de 40 anos de experiência. A Figura 4.4 mostra que a maioria dos participantes, 58.8% são funcionários do poder público e 17.6% são ativistas ou atuam em associações ligadas à mobilidade urbana. A Figura 4.5 mostra que 64.7% dos participantes são do gênero masculino e 35.3% do gênero feminino. Não houve participantes que declararam serem de outros gêneros.

As respostas às perguntas abertas trouxeram muitas ideias de funcionalidades para trabalhos futuros. As dificuldades no uso da ferramenta indicam que há possibilidade de melhorias a serem feitas na usabilidade e otimização do servidor. Os resultados mostram uma percepção majoritariamente positiva dos participantes na utilidade, na facilidade de uso e na intenção de uso. Dessas, a intenção de uso foi a melhor avaliada mostrando que as análises da BikeScienceWeb são úteis e que os usuários adotariam a ferramenta no seu dia-a-dia para análise da mobilidade urbana. As avaliações negativas citam dificuldade em interpretar o mapa, necessidade de mais informações sobre os fluxos dos quartis e de camadas, e por fim, lentidão.

Capítulo 5

Conclusão e Trabalhos Futuros

A ferramenta BikeScienceWeb é uma ferramenta de código aberto que surgiu da necessidade de facilitar o uso da BikeScience, dispensando a necessidade de conhecimentos em programação e ciência de dados. Além disso, por ser uma aplicação web está disponível para uso a partir de qualquer PC, independente do sistema operacional.

A avaliação do questionário mostrou que a ferramenta tem grande potencial para ser utilizada nas análises por especialistas da área de mobilidade urbana. A ferramenta foi lançada e já está disponível para uso. Inclusive, está sendo usada por especialistas em mobilidade urbana ativa da CET-SP, o órgão responsável pelo trânsito da maior cidade da América Latina.

A realização do trabalho teve uma grande curva aprendizado de novas tecnologias de desenvolvimento de sistemas de software. A organização e planejamento do desenvolvimento junto aos orientadores foi fundamental para a priorização eficaz das entregas com maior valor agregado para os usuários.

Um grande aprendizado foi que um melhor planejamento inicial da arquitetura do software teria facilitado o desenvolvimento. Mas ainda assim mudanças nos planos acontecem, logo, o sistema de software deve ter flexibilidade para receber alterações imprevistas ao longo do fluxo de desenvolvimento. Nesse aspecto, a utilização do arcabouço Vue.js se mostrou bastante adequada devido a sua flexibilidade e modularização via componentes.

Para trabalhos futuros, cito alguns itens que estavam na proposta inicial deste trabalho e também nas sugestões dos participantes da pesquisa, como a adição de novas funcionalidades como *upload* de *shapefiles*, exportação de mapas, gráficos e inclusão de rotas para bicicletas, inclusão de dados da Pesquisa OD 2007 para comparação, busca no mapa por endereço, comparação de mapas lado a lado e melhorias na visibilidade e usabilidade.

Referências

- [BATISTA *et al.* 2016] Daniel Macedo BATISTA, Alfredo GOLDMAN, Roberto Hirata JR. e Fabio KON. “Interscity: addressing future internet research challenges for smart cities”. Em: *7th IEEE International Conference on Network of the Future* (2016) (citado na pg. 3).
- [BROWN 2019] Simon BROWN. *Software Architecture for Developers - Visualise, document and explore your software architecture - Volume 2*. 2ª ed. Leanpub, set. de 2019 (citado na pg. 29).
- [BROWN 2021] Simon BROWN. *The C4 model for visualising software architecture*. Dez. de 2021. URL: <https://c4model.com/> (acesso em 11/12/2021) (citado na pg. 28).
- [BUEFY 2021] BUEFY. *Documentation*. Nov. de 2021. URL: <https://buefy.org/documentation/> (acesso em 24/11/2021) (citado na pg. 28).
- [CET-SP 2021a] CET-SP. *Dados abertos*. Dez. de 2021. URL: http://geosampa.prefeitura.sp.gov.br/PaginasPublicas/_SBC.aspx (acesso em 10/12/2021) (citado na pg. 17).
- [CET-SP 2021b] CET-SP. *Mapa de Infraestrutura Cicloviária*. Dez. de 2021. URL: <http://www.cetsp.com.br/consultas/bicicleta/mapa-de-infraestrutura-ciclovitaria.aspx> (acesso em 10/12/2021) (citado na pg. 17).
- [CHOI 2020] David CHOI. *Full-Stack React, TypeScript, and Node: Build cloud-ready web applications using React 17 with Hooks and GraphQL*. 1ª ed. Packt Publishing Ltd., dez. de 2020 (citado na pg. 7).
- [COMPOSE 2021a] Docker COMPOSE. *Documentation*. Nov. de 2021. URL: <https://docs.docker.com/compose/> (acesso em 24/11/2021) (citado na pg. 26).
- [COMPOSE 2021b] Docker COMPOSE. *Network Documentation*. Nov. de 2021. URL: <https://docs.docker.com/compose/networking/> (acesso em 24/11/2021) (citado na pg. 26).
- [DAVIS 1989] Fred D. DAVIS. “Perceived usefulness, perceived ease of use, and user acceptance of information technology”. Em: *MIS Q.* 13.3 (set. de 1989), pgs. 319–340. ISSN: 0276-7783. DOI: 10.2307/249008. URL: <https://doi.org/10.2307/249008> (citado na pg. 41).

- [DOCKER 2021] DOCKER. *Documentation*. Nov. de 2021. URL: <https://docs.docker.com/get-started/overview/> (acesso em 24/11/2021) (citado na pg. 26).
- [ESLINT 2021] ESLINT. *Website*. Nov. de 2021. URL: <https://eslint.org/> (acesso em 25/11/2021) (citado na pg. 28).
- [ESRI 2021] ESRI. *Shapefile Technical Description*. Dez. de 2021. URL: <https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf> (acesso em 10/12/2021) (citado na pg. 17).
- [FLASK 2021] FLASK. *Documentation*. Nov. de 2021. URL: <https://flask.palletsprojects.com/en/2.0.x/foreword/> (acesso em 24/11/2021) (citado na pg. 23).
- [GEOPANDAS 2021] GEOPANDAS. *Documentation*. Dez. de 2021. URL: <https://geopandas.org> (acesso em 13/12/2021) (citado na pg. 24).
- [INTERSCITY 2021] INTERSCITY. *Website*. Nov. de 2021. URL: <https://interscity.org/about/> (acesso em 28/11/2021) (citado na pg. 3).
- [JADHAV *et al.* 2015] Madhuri A JADHAV, Balkrishna R SAWANT e Anushree DESHMUKH. “Single page application using angularjs”. Em: *International Journal of Computer Science and Information Technologies* 6.3 (2015), pgs. 2876–2879 (citado na pg. 7).
- [JSON 2021] JSON. *Website*. Dez. de 2021. URL: <https://www.json.org/json-en.html> (acesso em 12/12/2021) (citado na pg. 27).
- [KON *et al.* 2020] Fabio KON *et al.* “Abstracting mobility flows from bike-sharing systems”. Em: *Public Transport* (2020) (citado nas pgs. 4, 15, 16, 18, 19).
- [LEAFLET 2021] LEAFLET. *Documentation*. Nov. de 2021. URL: <https://leafletjs.com/> (acesso em 25/11/2021) (citado na pg. 25).
- [MOZILLA 2021] MOZILLA. *Documentação*. Dez. de 2021. URL: <https://developer.mozilla.org/pt-BR/docs/Web/HTML> (acesso em 12/12/2021) (citado na pg. 27).
- [NPM 2021] NPM. *Website*. Dez. de 2021. URL: <https://www.npmjs.com/about> (acesso em 12/12/2021) (citado na pg. 27).
- [OD 2017a] Pesquisa OD. *Base de dados*. Dez. de 2017. URL: <https://transparencia.metrosp.com.br/dataset/pesquisa-origem-e-destino/resource/4362eaa3-c0aa-410a-a32b-37355c091075> (acesso em 02/12/2021) (citado nas pgs. 11, 12, 16, 17).
- [OD 2017b] Pesquisa OD. *Website*. Dez. de 2017. URL: <http://www.metro.sp.gov.br/pesquisa-od/responsabilidade.aspx> (acesso em 02/12/2021) (citado na pg. 9).
- [OD 2017c] Pesquisa OD. *Website*. Dez. de 2017. URL: <http://www.metro.sp.gov.br/pesquisa-od/informacoes-od.aspx> (acesso em 02/12/2021) (citado na pg. 9).

REFERÊNCIAS

- [OD 2017d] Pesquisa OD. *Website*. Dez. de 2017. URL: <http://www.metro.sp.gov.br/pesquisa-od/planejamento-preparacao.aspx> (acesso em 02/12/2021) (citado nas pgs. 9, 10).
- [OD 2019] Pesquisa OD. *Relatório Síntese*. Dez. de 2019. URL: <https://transparencia.metrosp.com.br/dataset/pesquisa-origem-e-destino/resource/b3d93105-f91e-43c6-b4c0-8d9c617a27fc> (acesso em 02/12/2021) (citado nas pgs. 8, 10, 11).
- [PANDAS 2021] PANDAS. *Documentation*. Dez. de 2021. URL: https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html (acesso em 13/12/2021) (citado na pg. 23).
- [PYTHON 2021] PYTHON. *Documentation*. Nov. de 2021. URL: <https://docs.python.org/3/faq/general.html#what-is-python> (acesso em 24/11/2021) (citado na pg. 23).
- [SOUZA 2018] Higor Amario de SOUZA. “Assessment of spectrum-based fault localization for practical use”. Tese de dout. São Paulo, Brasil: Instituto de Matemática e Estatística, abr. de 2018 (citado na pg. 41).
- [SOUZA *et al.* 2021] Higor Amario de SOUZA, Edison de OLIVEIRA VIANNA, Edlene Carneiro de SOUZA e Fabio KON. “Implantação e uso da ferramenta de análise de mobilidade de bicicletas BikeScience na CET: Identificando caminhos cicláveis em São Paulo”. Em: *Revista UniCET* (2021) (citado nas pgs. 4, 5, 9).
- [TANEJA e GUPTA 2014] Sheetal TANEJA e Pratibha R. GUPTA. “Python as a tool for web server application development”. Em: 2014 (citado na pg. 7).
- [TOASTR 2021] TOASTR. *Repository*. Nov. de 2021. URL: <https://github.com/CodeSeven/toastr> (acesso em 25/11/2021) (citado na pg. 28).
- [VÁZQUEZ-INGELMO *et al.* 2020] Andrea VÁZQUEZ-INGELMO, Alicia GARCÍA-HOLGADO e Francisco J. GARCÍA-PEÑALVO. “C4 model in a software engineering subject to ease the comprehension of uml and the software”. Em: *2020 IEEE Global Engineering Education Conference (EDUCON)*. 2020, pgs. 919–924. DOI: [10.1109/EDUCON45650.2020.9125335](https://doi.org/10.1109/EDUCON45650.2020.9125335) (citado na pg. 28).
- [VUE-TOASTR-2 2021] VUE-TOASTR-2. *Repository*. Nov. de 2021. URL: <https://github.com/saivarunk/vue-toastr-2> (acesso em 25/11/2021) (citado na pg. 28).
- [VUE2LEAFLET 2021] VUE2LEAFLET. *Documentation*. Nov. de 2021. URL: <https://vue2-leaflet.netlify.app/quickstart> (acesso em 25/11/2021) (citado na pg. 25).
- [VUEJS 2021] VUEJS. *Guide*. Nov. de 2021. URL: <https://vuejs.org/v2/guide/> (acesso em 24/11/2021) (citado na pg. 24).
- [VUEX 2021] VUEX. *Documentation*. Nov. de 2021. URL: <https://vuex.vuejs.org/#what-is-a-state-management-pattern> (acesso em 26/11/2021) (citado nas pgs. 24, 25).