

Colletivo: Uma Plataforma para a Gerência de Dados de Transporte Público no formato GTFS

Mateus Orlandin Lorenzatti¹, Felipe Martin Sampaio¹

¹ Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Farroupilha - Farroupilha, RS - Brasil

mateusolorenzatti@gmail.com, felipe.sampaio@farroupilha.ifrs.edu.br

Abstract. *The availability of updated data on public transport plays a key role in encouraging the use of these means, thus promoting environmental, social benefits and urban mobility. Considering this need, which is more latent in smaller cities, ByBus was created, a project that united private and public partnerships to map and make public transportation data available in the city of Farroupilha-RS. The data were organized in the GTFS format, standard used worldwide in urban mobility applications. After the completion of the project, needs were observed and so the creation of Colletivo, a free and open-source platform for the management of GTFS data, was made possible. The idea is that the platform accompanies the process of digitalization of public transport data, from its insertion via GPS or manual, to its publication in the most widely used applications worldwide. This paper presents the technologies and architecture used to build the system, as well as the results obtained. The tool allows the manual insertion of route data, connecting the points of the existing stops, calculating the route between the points and saving the necessary information in a database structured along the lines of GTFS.*

Resumo. *A disponibilização de dados atualizados sobre o transporte público desempenha um papel fundamental ao incentivar o uso desses meios, promovendo assim benefícios sociais, de meio ambiente e mobilidade urbana. Tendo em vista essa necessidade, que é mais latente em cidades menores, surgiu o ByBus, projeto que uniu parcerias privadas e públicas para mapear e disponibilizar os dados de transporte público na cidade de Farroupilha-RS. Os dados foram organizados no formato GTFS, padrão utilizado mundialmente nos aplicativos de mobilidade urbana. Após a finalização do projeto, necessidades foram observadas e assim se oportunizou a criação do Colletivo, uma plataforma gratuita e open-source para a gestão de dados GTFS. A ideia é de que a plataforma acompanhe o processo de digitalização dos dados de transporte público, desde a sua inserção via GPS ou manual, até a sua publicação nos aplicativos mais utilizados mundialmente. Neste trabalho são apresentadas as tecnologias e a arquitetura pensada para a construção do sistema, bem como os resultados obtidos. A ferramenta permite a inserção manual dos dados de trajetos, ligando os pontos das paradas existentes, calculando o trajeto entre os pontos e gravando as informações necessárias em uma base de dados estruturada nos moldes do GTFS.*

1. Introdução

O registro e a exibição de dados do transporte público coletivo são fatores muito importantes para a mobilidade e a acessibilidade nas cidades. Estas necessidades são ainda mais evidentes em municípios menores, onde dificilmente há um sistema intuitivo e prático para se consultar informações de itinerários, como é o caso do município de Farroupilha-RS. Pensando nessas demandas da cidade, no ano de 2017 o município de Farroupilha participou do projeto “Feito na Biblioteca”, organizado pela Organização Não-Governamental norte-americana Caravan Studios (TECHSOUP, 2023) em parceria com a Biblioteca Pública Olavo Bilac. O projeto consistiu em apresentar ideias de soluções tecnológicas nas bibliotecas públicas de diferentes regiões do Brasil para que a comunidade compareça e vote nas demandas que julgasse mais importantes, dentro do contexto em que viviam. A ideia vencedora do projeto em Farroupilha foi o “ByBus”, que consiste em apresentar os dados dos itinerários de forma dinâmica através de um aplicativo mobile (CARAVAN STUDIOS, 2023).

O projeto ByBus ganhou vida com o auxílio de servidores e alunos do IFRS Campus Farroupilha, que voluntariamente compareceram às rotas com dispositivos para coletar dados dos horários e pontos de parada dos ônibus da cidade, juntamente com a empresa Trillium Transit (TRILLIUM TRANSIT, 2023), que auxiliou na gerência e na coleta dos dados (IFRS, 2018). Estas informações foram organizadas e disponibilizadas à comunidade por meio do aplicativo Transit (TRANSIT, 2023), onde é possível consultar as rotas do transporte público por meio de dispositivos móveis.

Os dados que foram processados e enviados ao Transit foram estruturados no formato GTFS. O GTFS, sigla que significa *General Transit Feed Specification*, que pode ser traduzido para “Especificação Geral de Feed de Transporte Público”, é uma padronização para que agências de transporte público possam disponibilizar as informações a serem consumidas por diversas aplicações em larga escala (MOBILITYDATA, 2023). Mais informações sobre o GTFS serão exibidas na Seção 2.1.

Esses dados foram processados, organizados e exportados utilizando a ferramenta paga GTFS Manager, software proprietário mantido pela própria Trillium Transit. Com o fim do projeto ByBus, ajustes nos trajetos, novas rotas ou paradas não puderam mais ser feitos, e se faria necessário o acesso à uma nova ferramenta que suportasse essas necessidades.

Conhecendo esse contexto, o atual trabalho realizou o desenvolvimento da plataforma “Colletivo”, um sistema web que permite a inserção, edição, manutenção, publicação e controle de dados de trânsito no formato GTFS. A ideia é que a plataforma seja *open-source* (Código Aberto) e gratuita, para que tenha a contribuição da comunidade e que possa ser oferecida para cidades menores, onde muitas vezes os orçamentos podem desencorajar a contratação de ferramentas para gestão destes dados, como o caso do GTFS Manager. O público alvo da plataforma são os administradores das operadoras do transporte público e outros interessados, como funcionários das prefeituras e/ou secretarias municipais de trânsito, por exemplo.

O Colletivo pretende unificar os recursos supracitados de forma a cumprir com os seguintes objetivos:

01. Estruturar uma base de dados que comporte o formato GTFS.
02. Disponibilizar uma API para exibir os dados das entidades GTFS da organização.
03. Importar os dados de rotas coletados por meio de aplicativos mobile ou aparelhos de GPS e transformar automaticamente em GTFS.
04. Inserir os dados de rotas com interação em tela por meio de mapas.
05. Permitir que o usuário tenha controle sobre a visibilidade de rotas e paradas por parte de terceiros.
06. Expor dados das rotas e trajetos disponíveis de forma aberta para que seja consultado por diferentes aplicações de terceiros.
07. Oferecer uma exibição geral dos dados existentes na base da agência sendo gerida.

Este artigo apresentará os resultados obtidos no desenvolvimento de alguns dos objetivos mencionados. Os objetivos 01, 02, 04 e 06 foram completamente contemplados. Já os objetivos 03, 05 e 07 não foram totalmente contemplados e, desta forma, serão apontados como perspectivas futuras a este trabalho.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta o referencial teórico, contando com conceitos fundamentais do projeto e uma análise de soluções do mercado voltadas à essa pauta. Na sequência, a Seção 3 mostra a metodologia, com as escolhas de tecnologia e arquitetura para a implantação do Coletivo. Logo após, na Seção 4, são apresentados os resultados obtidos com o desenvolvimento. E, por fim, as considerações finais são apresentadas na Seção 5.

2. Referencial Teórico

Segundo Rodrigues e Sorratini (2008) o transporte coletivo urbano possui grande importância na configuração de trânsito que interliga as regiões das cidades, contribuindo com problemas como congestionamentos, acidentes de trânsito e impactos ambientais. Além disso, o transporte coletivo apresenta uma importância social para a população:

O transporte coletivo tem importância fundamental dentro do contexto geral do transporte urbano, na medida em que é essencial para a população de baixa renda e, ao mesmo tempo, uma importante alternativa a ser utilizada como estratégia para redução das viagens por automóvel, contribuindo para a redução dos congestionamentos, da poluição ambiental, dos acidentes de trânsito e do consumo de combustível. (RODRIGUES E SORRATINI, 2008)

O estudo de Rodrigues e Sorratini (2008) ainda apresenta uma pesquisa realizada, que buscou identificar a satisfação de usuários de transporte público em diferentes indicadores. É possível perceber que o indicador “Sistema de informação - Informações sobre linhas e horários nos pontos e dentro dos ônibus” apresentou uma insatisfação de 75%.

Visando reduzir a insatisfação dos usuários sobre as informações disponíveis dos itinerários nas cidades, surgiu o GTFS, que será apresentado em seguida, na seção 2.1. E junto com o GTFS, soluções de gerência dos dados nesse formato surgiram no mercado, e algumas delas serão apresentadas na seção 2.2.

2.1. GTFS

O formato GTFS foi concebido no ano de 2005, por Bibiana McHugh, então gerente de TI na empresa TriMet, uma agência de transporte público da metrópole de Portland, no estado do Oregon, nos Estados Unidos. Bibiana, naquele mesmo ano, fez uma viagem e se frustrou com a dificuldade em conseguir informações sobre rotas de ônibus em uma cidade desconhecida. Segundo ela, naquele momento era mais favorável utilizar aplicativos de GPS (Do inglês *Global Positioning System*, que pode ser traduzido como Sistema de Posicionamento Global), encorajando os usuários a usarem carros ao invés de dar preferência ao transporte público (MCHUGH, 2013).

Após contatar a empresa Google sobre a ideia de padronizar os dados de transporte público em um formato simples e universal, foi firmada uma parceria com a TriMet para a criação de uma primeira versão do GTFS. Assim foi criada a ferramenta Google Transit, que é utilizada para que agências de transporte público possam alimentar os dados a serem disponibilizados no aplicativo Google Maps (MCHUGH, 2013).

O formato GTFS propõe uma estrutura de entidades e atributos relacionando informações como paradas, rotas, horários, formatos e trajetos, entre outros elementos relevantes. Esses dados podem ser armazenados em arquivos de texto, maneira comumente usada na exportação para aplicativos de transporte, mas também podem ser armazenados em bancos de dados relacionais, pois o formato estabelece relações de cardinalidade entre as entidades. A Figura 1 mostra um diagrama geral com as relações hierárquicas entre cada entidade do GTFS. É interessante pontuar que existem entidades obrigatórias, que são a base da estruturação do formato, mas também existem entidades opcionais, permitindo conter informações complementares (FORTIN; MORENCY; TRÉPANIE, 2016, p. 21).

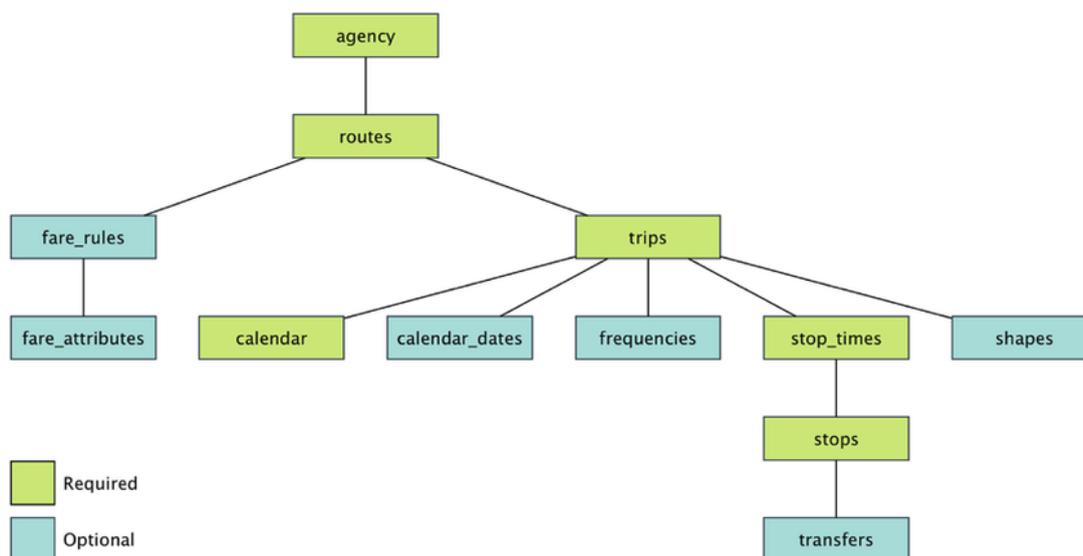


Figura 1. Representação estrutural do GTFS.

Fonte: Fortin, Morency e Trépanie (2016)

Dentro dessa relação, cada entidade tem as suas características que compõem o GTFS. Abaixo temos uma explicação mais detalhada para as principais entidades do formato:

- *Agency* (Agência): No topo da hierarquia está a entidade “Agency”, do inglês agência. Essa entidade é encarregada de representar os dados da empresa responsável pela operação das rotas cadastradas.
- *Routes* (Rotas): Ela representa em alto nível de abstração uma rota de ônibus, caracterizada por ir de ponto A a B.
- *Trips* (Trajetos): O trajeto é a representação mais concreta da rota, já que se relaciona com o maior número de entidades, caracterizando informações de horários, formatos e frequência. Pode conter informações como suporte ao embarque de cadeirantes e de ciclistas.
- *Stops* (Paradas): É a entidade com as informações relevantes sobre uma parada de ônibus, contendo principalmente as coordenadas geográficas e um nome para a identificação. Pode conter informações como acessibilidade para cadeirantes, por exemplo.
- *Stop_Times* (Paradas-Tempo): É a relação feita entre os trajetos e as paradas. Irá conter como campo principal o horário (podendo ser de chegada ou partida) e a ordem da parada dentro da sequência em que o trajeto percorre.
- *Shapes* (Formatos): Não é uma entidade obrigatória, porém se mostra muito válida para a exibição dos trajetos em mapas. Essa entidade representa cada coordenada geográfica por onde o veículo percorre no trajeto. Assim, a sequência das coordenadas irá permitir a exibição de uma linha contínua pelas ruas onde o veículo passa.
- *Calendar* (Calendário): Representa a disponibilidade do trajeto em cada dia da semana. Possui um campo binário para cada dia da semana, indicando se o dia está disponível ou não.

2.2. Soluções Relacionadas

Existem diversas soluções no mercado que permitem o gerenciamento de dados GTFS, contando com diferentes funcionalidades e custos. Nesta seção, serão apresentadas três aplicações que serviram de base para o desenvolvimento do Colletivo: GTFS Manager, Moovit Transit Data Manager e Mobilibus.

Como mencionado anteriormente, o GTFS Manager (TRILLIUM TRANSIT, 2023), da norte-americana Trillium Transit, foi a ferramenta utilizada durante a execução do projeto ByBus. Essa ferramenta oferece uma variedade de recursos para organizar as rotas, permitindo o controle dos trajetos existentes e seus horários, bem como a exibição de gráficos comparativos dos diferentes horários e momentos em que os ônibus passam pelas paradas. No entanto, uma observação feita durante a utilização do GTFS Manager no ByBus é que a ferramenta não oferece a funcionalidade de importação direta de dados coletados por GPS, o que significa que as paradas precisam ser adicionadas (ou reutilizadas) e conectadas manualmente.

O Moovit Transit Data Manager (MOOVIT, 2023), desenvolvido pela renomada empresa israelense Moovit, é considerado uma das opções mais abrangentes e eficientes disponíveis no mercado. Essa solução oferece a possibilidade de gerenciar e controlar os dados de GTFS, permitindo uma integração com informações em tempo real. Além disso, graças às soluções abrangentes da Moovit, conhecidas como MaaS (*Mobility as a Service* - Mobilidade como Serviço), todo o ecossistema oferece uma ampla gama de recursos, especialmente direcionados para as áreas metropolitanas que utilizam diferentes modos de transporte, como metrô, por exemplo.

Uma solução brasileira presente no mercado é a Mobilibus (MOBILIBUS, 2023). A ferramenta oferece como diferencial a integração por meio de diferentes fontes para a inserção de rotas, como telemetria, GPS, bilhetagem eletrônica ou integração de pagamento. Além disso, um outro diferencial citado pela empresa é a oferta de planos mais acessíveis, permitindo uma maior viabilidade para a implantação. Entretanto, a solução ainda disponibiliza um aplicativo mobile a ser utilizado por usuários finais para a consulta dos trajetos e navegação. Isso demonstra um foco ligeiramente diferente da proposta do Coletivo, que é de exportar dados para aplicativos amplamente utilizados e consolidados, já disponíveis em smartphones.

3. Metodologia

A metodologia de desenvolvimento escolhida foi o Scrum, estabelecendo ciclos de desenvolvimento e revisão dos requisitos nas Sprints, etapas de uma a duas semanas onde vão sendo definidas as tarefas a serem concluídas em cada etapa. A cada sprint nova, foi definido o próximo backlog, ou a lista de itens a serem concluídos (SOMMERVILLE, 2011).

O desenvolvimento da aplicação foi dividido em duas partes: (1) a parte responsável pelo controle dos acessos e dados por parte do servidor, também conhecida como *back-end*; e (2) a parte que apresenta os dados e controla as interações em tela, o *front-end*. A linguagem escolhida para ambas as partes é o TypeScript (MICROSOFT, 2023), uma variação da linguagem JavaScript com recursos de tipagem e orientação a objetos. Criada em 2012 e mantida pela Microsoft, o TypeScript é uma linguagem que é automaticamente "traduzida" para o JavaScript, permitindo que os programas rodem em navegadores ou em engines como o NodeJS.

O NodeJS é uma plataforma de desenvolvimento e execução assíncrona que permite a construção de aplicações escaláveis (THE OPENJS..., 2023). A plataforma utiliza a linguagem JavaScript como base para a codificação. Por ser versátil, o NodeJS permite o desenvolvimento tanto de aplicações *front-end* quanto *back-end*.

A comunicação entre as partes é feita por meio do conceito de API REST . O REST é um padrão comumente usado por serviços web, baseado no protocolo HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto). Por estabelecer comunicação entre aplicações, o REST é considerado uma API (*Application Programming Interface* - Interface de Programação de Aplicações). O cliente faz a requisição para o servidor, detalhando a ação que deseja fazer por meio de um verbo, que poderá indicar consulta, alteração, criação ou exclusão de dados, entre outras ações.

O servidor recebe a requisição, executa a tarefa solicitada e retorna ao cliente uma mensagem, podendo ser de erro caso algo não tenha ocorrido conforme o esperado (RED HAT INC., 2020).

3.1. Back-end

O *back-end* foi desenvolvido utilizando o framework NestJS. O framework foi criado em 2017 por Kamil Mysliwicz, e apresenta recursos como padronização e estruturação que facilita a organização do código. Um de seus princípios é o CoC, *Convention Over Configuration* (do inglês "Convenção Sobre Configuração"), que é a predefinição de arquivos e diretórios que tornam mais intuitiva a disposição de lógicas (MYSLIWIEC, 2022). Por conta dessas características e da boa performance na engine NodeJS, esse foi o framework escolhido para o desenvolvimento.

Os dados são armazenados no banco de dados Postgres (POSTGRESQL, 2023). O Postgres é um dos mais populares SGBDs (Sigla para *Data Base Management System* ou Sistema de Gerenciamento de Banco de Dados) no mercado, conhecido pela sua robustez, por ser gratuito e de código aberto. É um banco de dados relacional, criado no ano de 1986 na University of California e sofre atualizações e aprimoramentos até os dias atuais.

Uma biblioteca importante para o funcionamento do *back-end* é o TypeORM (TYPEORM, 2023). Essa biblioteca desempenha um papel essencial na interação entre o código e as entidades do banco de dados. Baseada no conceito de ORM (*Object-Relational Mapping* - Mapeamento Objeto-Relacional), o TypeORM representa as entidades do banco de dados como objetos, facilitando os acessos e alterações por meio de uma camada de controle, eliminando a necessidade de utilizar comandos SQL (Sigla para *Structured Query Language* - Linguagem de Consulta Estruturada) diretamente.

Dentre as principais vantagens do uso de um ORM, destacam-se a agilidade e produtividade no desenvolvimento, uma vez que o TypeORM simplifica a manipulação dos dados e abstrai a complexidade das consultas SQL. Além disso, a biblioteca oferece recursos de segurança para acesso e alteração dos dados, protegendo contra vulnerabilidades e proporciona portabilidade entre diferentes bancos de dados (TORRES et al., 2017).

A estratégia de autenticação utilizada foi a Bearer, também conhecida como autenticação por token. Como o REST não armazena a sessão do usuário, a forma de identificar o acesso e saber se há permissão para acessar o recurso é por meio da geração de um token quando o cliente faz o login. Uma vez gerado o token, será indicado o tempo de validade, e até lá o cliente pode usar para fazer as demais requisições que desejar (SMARTBEAR, 2023).

O tipo de token escolhido foi o JWT (JSON Web Token), que é um dos padrões mais utilizados no mercado. Ele leva em seu conteúdo, de forma criptografada, dados como ID ou nome de usuário e uma assinatura. Essa assinatura é baseada em uma chave que é de propriedade do servidor, indicando que o servidor poderá saber qual assinatura foi gerada por ele e assim validar a identidade de quem fez a requisição (AUTH0,

2023).

3.2. Front-end

O *front-end* foi desenvolvido utilizando o framework Angular (GOOGLE, 2023). Com a estrutura muito similar ao do NestJS, o Angular é um framework mantido pela empresa Google é uma das opções mais utilizadas no mercado para *Single-Page Applications*, do inglês Aplicações de Página Única. As SPAs são páginas que gerenciam a própria navegação, sem a necessidade de contatar o servidor a cada tela acessada, melhorando a experiência dos usuários tanto em computadores quanto em smartphones (SCHNEIDER, 2016).

Um dos principais elementos gráficos a serem exibidos ao usuário em tela são os mapas. Para isso, a biblioteca escolhida foi a Mapbox (MAPBOX, 2023). A escolha se baseou na possibilidade de utilizar um plano gratuito que ofereça a exibição de mapas e a geração das rotas entre pontos no mapa. Além disso, a biblioteca tem um grande apoio da comunidade, o que propicia uma agilidade em resolver possíveis problemas e tirar dúvidas.

As informações a serem exibidas nos mapas foram parametrizadas em um formato específico para a exibição de dados geoespaciais: GeoJSON (BUTLER et al., 2016). Baseado no JSON (*JavaScript Object Notation* - Notação de Objeto do JavaScript), o formato propõe alguns padrões para a projeção de elementos como pontos, linhas, polígonos, entre outros. Os elementos são baseados nas coordenadas geográficas dos pontos nos mapas, compostas por longitude (medida em graus até o Meridiano de Greenwich) e latitude (medida em graus até a Linha do Equador).

Outro recurso do Mapbox que é imprescindível para o projeto é o cálculo das rotas entre os pontos. Para isso, foi utilizada a Directions API. A API funciona recebendo as coordenadas entre os dois pontos desejados e retorna a sequência de coordenadas para serem seguidas a fim de se chegar do ponto A ao ponto B, respeitando todos os fluxos do trajeto. A API ainda pode ser parametrizada para fornecer dados de trajetos de carro, bicicleta e a pé.

Já para os componentes visuais como campos de texto, estrutura de página e botões, a biblioteca escolhida foi o Bootstrap (BOOTSTRAP, 2023). O Bootstrap oferece uma gama de classes de CSS (*Cascading Style Sheets* - Folha de Estilos em Cascata) que agilizam e padronizam os elementos visuais contribuindo para um desenvolvimento mais eficiente e uma melhor experiência por parte dos usuários.

3.4. Ferramentas Utilizadas

O controle de versionamento do código foi feito utilizando a ferramenta Git (GIT, 2023), amplamente utilizada para tal função. Os códigos fonte desenvolvidos foram disponibilizados de forma pública na plataforma GitHub (GITHUB, 2023).

O ambiente de desenvolvimento utilizado para o desenvolvimento foi o Visual Studio Code (MICROSOFT, 2023). A escolha se deu por conta da integração nativa com a linguagem TypeScript, além da flexibilidade para a instalação de diversos plugins

que contribuem com o desenvolvimento.

Para executar os testes com a API do *back-end* foi utilizada a ferramenta Postman (POSTMAN, 2023). O Postman se faz útil por oferecer uma interface intuitiva e amigável que facilita a construção e o envio de solicitações HTTP para a API. É possível enviar diversos verbos, conteúdos, parâmetros, cabeçalhos, servindo até como forma de documentação, orientando sobre a forma em que as chamadas devem acontecer.

Já para monitorar o funcionamento do banco de dados, a ferramenta escolhida foi o DBeaver Community (DBEAVER, 2023). O DBeaver é uma ferramenta universal de acesso a bancos de dados. Permite conexões com os mais diversos tipos, seja SQL ou NoSQL. A versão utilizada no projeto é a Community, que é gratuita e *open-source*.

3.5. Visão Geral

Tendo como base os tópicos já apresentados, a visão geral do projeto pode ser visualizada na Figura 2, resumindo as principais linguagens, ferramentas e tecnologias adotadas em cada parte do desenvolvimento da aplicação Colletivo.

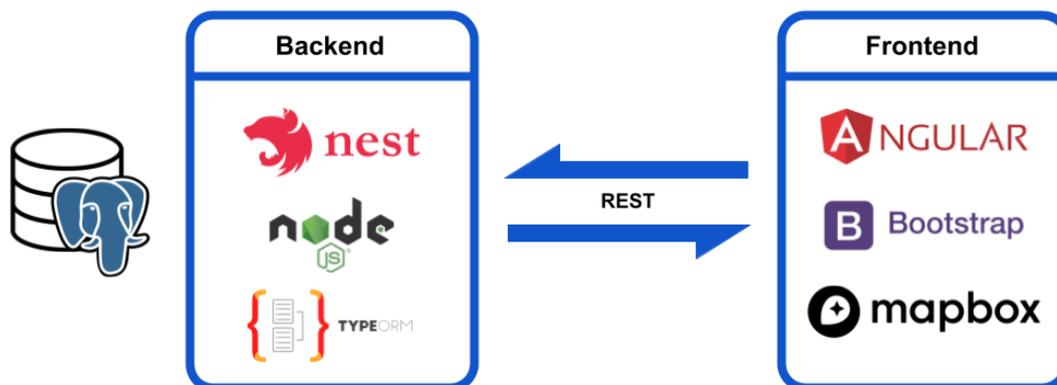


Figura 2. Diagrama representando a arquitetura do projeto, com as respectivas bibliotecas utilizadas em ambas as partes.

Fonte: Autoria própria.

4. Resultados

Nesta seção serão apresentados os resultados obtidos no desenvolvimento do Colletivo, sendo apresentado primeiramente a descrição do *back-end*, seguido pelas interfaces desenvolvidas para o *front-end*.

4.1. Back-end

A primeira etapa do desenvolvimento foi a criação das entidades por meio do TypeORM. Assim, foi gerada a base de dados sem utilizar diretamente comandos SQL. As entidades do GTFS criadas foram: Agency (Agência), Route (Rota), Trip (Trajeto), Shape (Formato), Stop-Time (Parada-Horário), e Stop (Parada). Além disso, foi criada

uma entidade User, para ter o controle de acesso à plataforma e possibilitar o login. O diagrama Entidade-Relacionamento que representa as tabelas criadas e sua hierarquia pode ser visualizado na Figura 3.

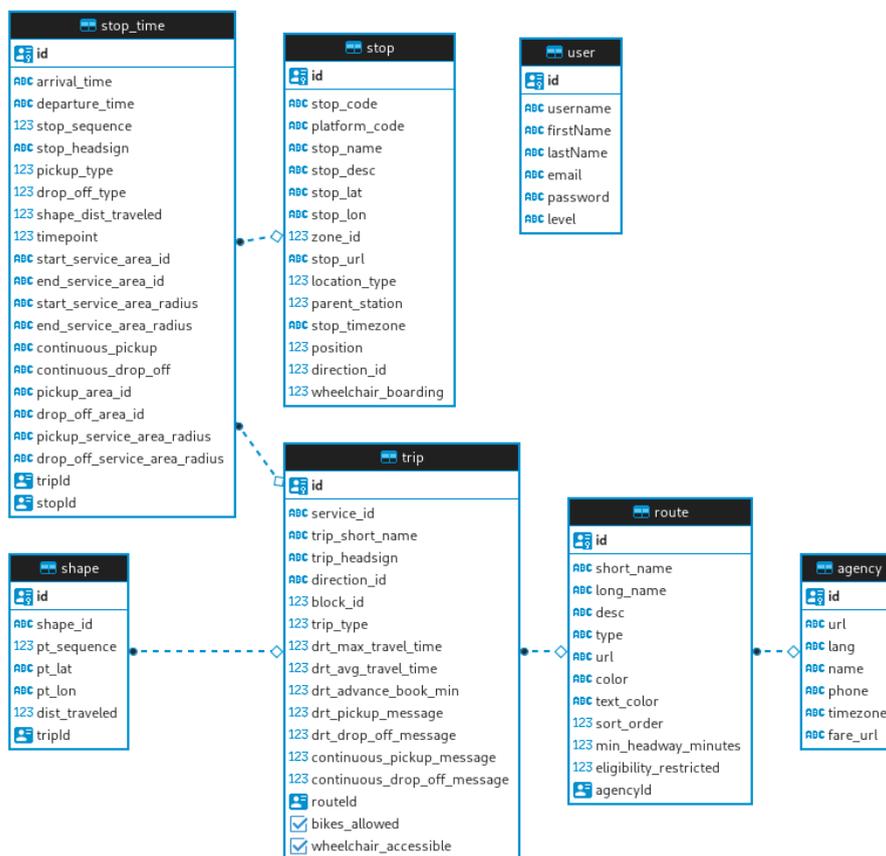


Figura 3. Diagrama ER da base de dados da aplicação.

Fonte: Autoria própria, gerado pela ferramenta DBeaver.

Para cada entidade do GTFS foram criados os métodos de: consulta (GET), inserção (POST), alteração (PUT) e exclusão (DELETE). Visando fornecer os dados de métodos disponíveis e quais as terminações necessárias para acessar cada entidade pelo cliente, foi adicionada uma documentação utilizando o módulo Swagger.

Com o uso do Swagger é possível visualizar todas as opções disponíveis a serem utilizadas na API do *back-end*. As rotas disponíveis aparecem divididas em suas *tags*, sendo uma reservada para as entradas relacionadas a autenticação e criação de usuário e as demais para cada entidade disponível no projeto. Esta visualização é útil para que desenvolvedores de aplicações terceiras possam saber quais as rotas e verbos disponíveis e os parâmetros necessários. A Figura 4 apresenta a página de swagger, que foi gerada automaticamente no NestJS.

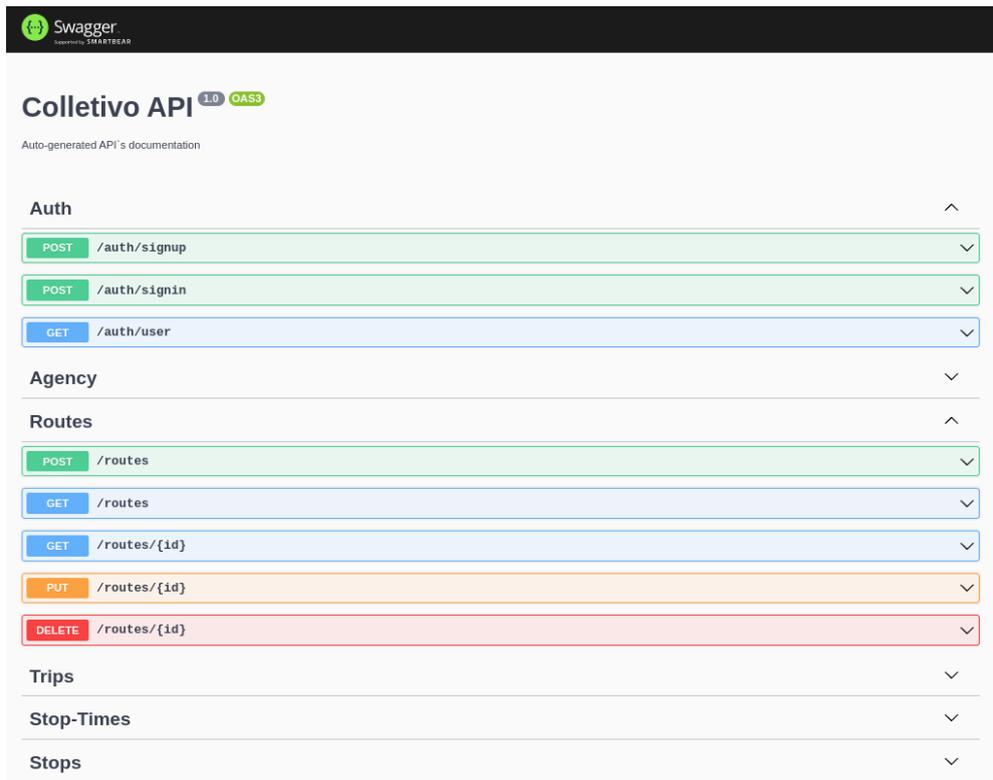


Figura 4. Página do Swagger, apresentando todas as rotas disponíveis na API.

Fonte: Autoria própria.

O Quadro 1 apresenta um exemplo de uma consulta de um trajeto, juntamente com a URL utilizada para fazer a consulta e o corpo da resposta no formato JSON retornado pela API. No exemplo mostrado, foram omitidos alguns registros para facilitar o entendimento. Os dados retornados pelo trajeto, além dos seus próprios atributos, também contam com a sua rota atrelada e também a lista de todas as suas paradas-tempo. Nos quadros aqui exibidos, estará sendo usado um host fictício chamado “colletivo.com” para poder exemplificar o uso da aplicação com disponibilidade na internet. Até o momento da escrita deste artigo, a aplicação foi executada e testada apenas em ambiente local (*localhost*).

GET | https://colletivo.com/trips/p_176923

```
{
  "id": "780cdce4-e07e-4ef8-92bd-a16264022d11",
  "service_id": "p_176923",
  "trip_short_name": "t_211484_b_6309_tn_0",
  "trip_headsign": "Belvedere x Cinquentenário x Praça",
  "route": {
    "id": "3eb713ea-e4a7-4322-a31c-6fc475de984d",
    "short_name": "Belvedere x Cinquentenário x Praça",
    "long_name": "Belvedere x Cinquentenário x Praça (9583)",
```

```

    "desc": "Belvedere x Cinquentenário x Praça (9583)"
  },
  "stop_times": [
    ( ... )
    {
      "id": "a623080b-89eb-40c6-87c5-ebf054d162ff",
      "arrival_time": "12:43:10",
      "departure_time": "12:43:10",
      "stop_sequence": 8,
      "stop_headsign": "",
      "stop_code": "2429642"
    },
    ( ... )
  ]
}

```

Quadro 1. Requisição de um trajeto e o retorno fornecido pela API.

Fonte: Autoria própria.

Um outro exemplo de consulta é a de paradas, conforme pode ser visto no Quadro 2. Essa consulta é feita da mesma forma, por meio do verbo GET, que irá retornar os dados referentes à parada solicitada.

```

GET | https://colletivo.com/stops/2429642

{
  "id": "8457735f-5204-48c7-a99d-35472c0bc074",
  "stop_code": "2429642",
  "platform_code": null,
  "stop_name": "Rodovia dos Romeiros com São Vicente (IFRS- Campus Farroupilha)",
  "stop_desc": null,
  "stop_lat": "-29.20198952",
  "stop_lon": "-51.3483274",
  "stop_timezone": "America/Sao_Paulo",
  "wheelchair_boarding": null
}

```

Quadro 2. Requisição de uma parada e o retorno fornecido pela API.

Fonte: Autoria própria.

4.2. Front-end

Tendo as rotas definidas no *back-end*, iniciou-se o desenvolvimento do *front-end*. A aplicação apresenta uma interface inicial de login, como pode ser visto na Figura 5. A tela também permite que o usuário crie um cadastro para utilizar a aplicação.

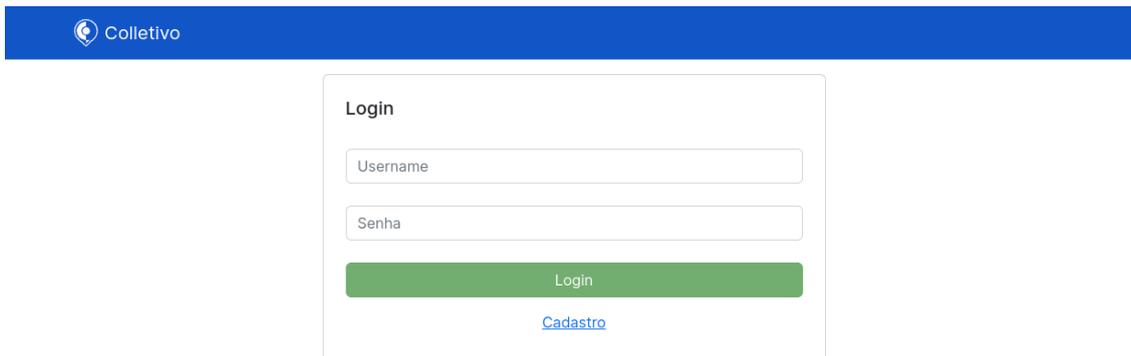


Figura 5. Interface de login da aplicação.

Fonte: Autoria própria.

Tendo o login efetuado com sucesso, o usuário é levado à interface central da aplicação, onde é possível visualizar as ações disponíveis. No momento de escrita deste artigo, os botões que estão ativados são “Criar Rota” e “Trajetos”, enquanto os demais estão desabilitados pois são funcionalidades que não foram desenvolvidas de forma completa. Essa tela pode ser visualizada na Figura 6.



Figura 6. Interface de home da aplicação.

Fonte: Autoria própria.

Ao clicar em “Criar Rota”, a aplicação mostra uma tela (Figura 7) onde é possível visualizar, na parte superior, um mapa com alguns pontos marcados. Esses pontos são as paradas disponíveis na cidade de Farroupilha, proveniente dos dados coletados pelo projeto ByBus. Para essa demonstração apresentada na Figura 7, não são todas as paradas que são mostradas em tela para não causar lentidão na utilização, visto que são mais de 1.000 (mil) paradas na base de dados.

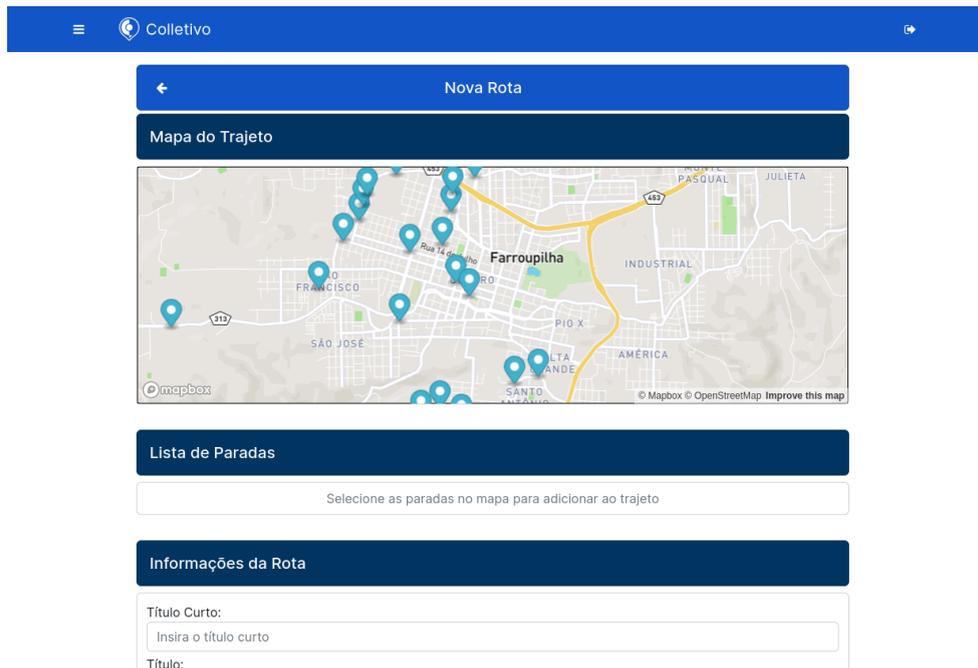


Figura 7. Interface para criação de uma nova rota.

Fonte: Autoria própria.

Nesse momento, o usuário pode ir clicando nas paradas conforme desejar e o trajeto vai sendo construído. As paradas vão sendo adicionadas na seção “Lista de Paradas” na ordem em que foram adicionadas. A Figura 8 apresenta essa lista criada, bem como o trajeto sendo traçado no mapa. Cada item da lista conterà um campo com o horário da parada, onde poderá ser informado o horário em que a rota passa pela parada.

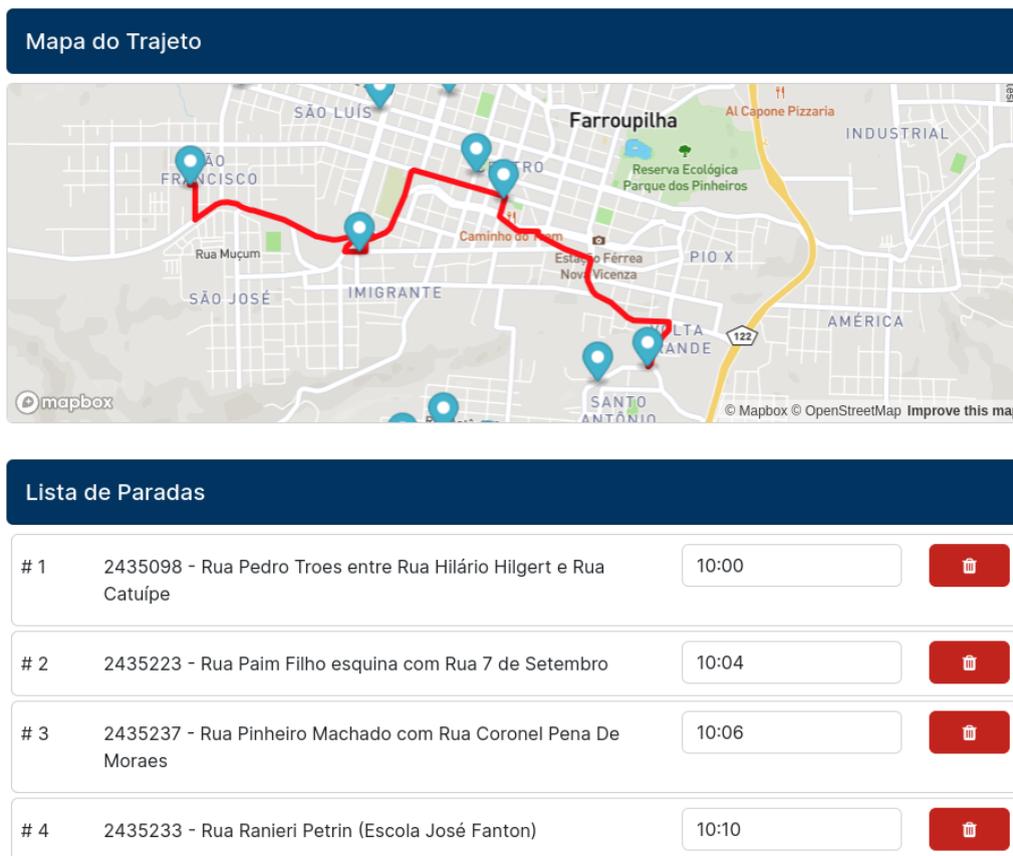


Figura 8. Paradas adicionadas na criação do trajeto.

Fonte: Autoria própria.

Caso o usuário adicione uma parada indesejada, é possível tirar ela do trajeto pelo botão com um ícone de lixeira no item da parada. Se a parada fica no início ou final do trajeto, ela apenas é removida, junto com o caminho que leva até ela. Já para os casos em que a parada estiver entre duas outras paradas, além de removida, a rota entre as outras duas é recalculada. Tal comportamento pode ser visto na Figura 9, em comparação com a Figura 8, onde a parada 3 foi removida e o caminho foi recalculado.

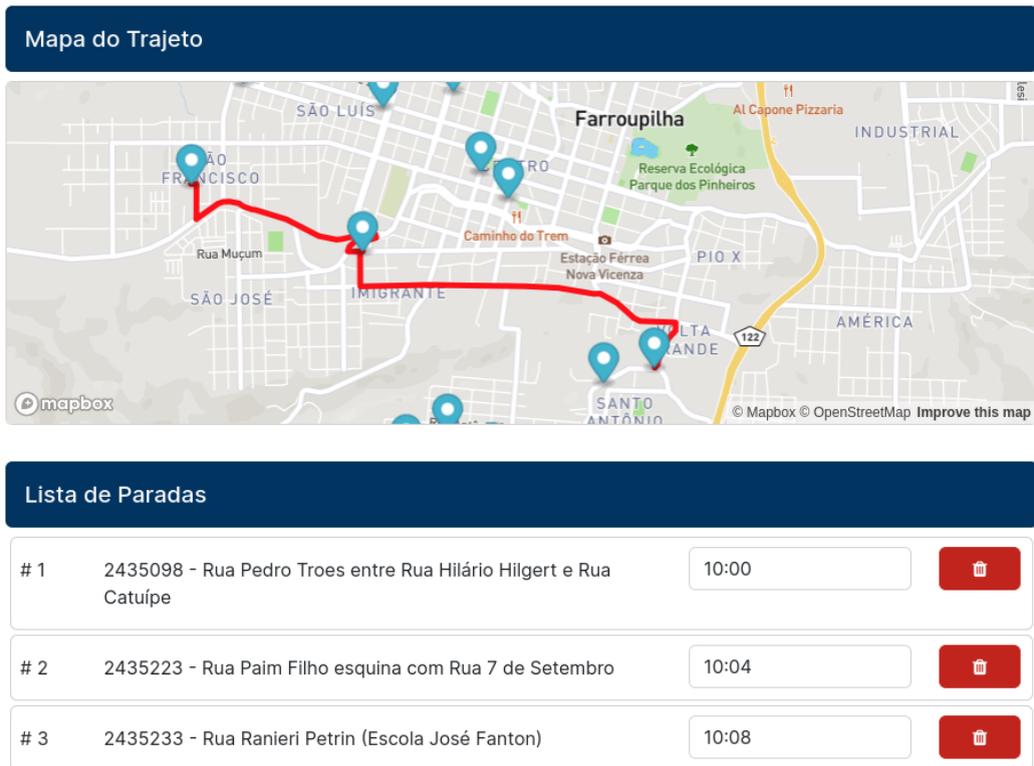


Figura 9. Removida a Parada 3 anterior, o trajeto foi recalculado.

Fonte: Autoria própria.

Após confirmar os dados de horários em cada parada no trajeto, o usuário precisa preencher as informações gerais do trajeto, como título, descrição, identificador e a agência a qual ele fará parte. A Figura 10 detalha essa parte da criação da rota.

Informações da Rota

Título Curto:
São Francisco x Volta Grande

Título:
São Francisco x Volta Grande (56)

Descrição:
São Francisco x Volta Grande (Rota 56). Horário de Partida: 10:00

ID de Serviço:
r_56

Agência:
Agência IFRS

[Voltar](#) [Criar Rota](#)

Figura 10. Preenchimento das informações gerais da rota.

Fonte: Autoria própria.

Tendo as informações confirmadas, o usuário segue em “Criar Rota”. Nesse momento, a aplicação cria os registros nas entidades do GTFS: Route, Trip, Shapes (com os dados vindos da Directions API) e Stop-Time (Vinculando as paradas com o trajeto). Depois que há a confirmação de que foram criados com sucesso, a aplicação é redirecionada à interface de visualização dos dados da rota, conforme apresentado na Figura 11.

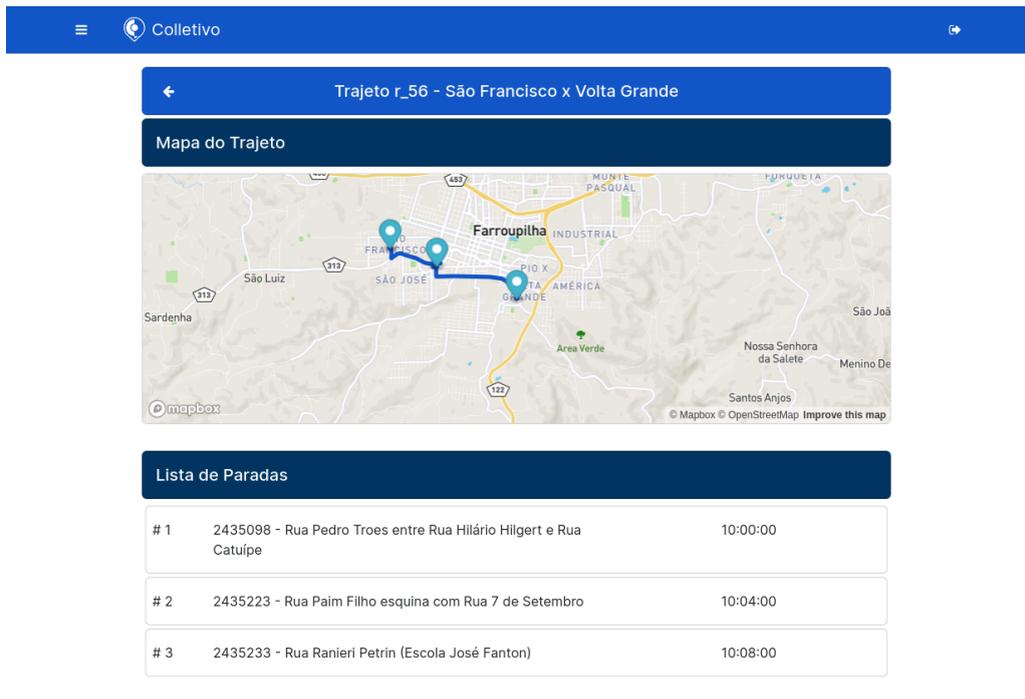


Figura 11. Interface de exibição dos dados da rota criada.

Fonte: Autoria própria.

Uma vez que o novo trajeto é criado, este irá aparecer na lista dos trajetos disponíveis, que pode ser acessado pelo botão “Trajetos” na página inicial. A Figura 12 apresenta a lista com os trajetos criados na base.



Figura 12. Interface de exibição da lista de Trajetos.

Fonte: Autoria própria.

Dessa forma o usuário pode visualizar os dados do trajeto que desejar. A Figura 13 mostra a interface com a mesma visualização da Figura 11, mas nesse caso está sendo apresentado um trajeto real, resultado dos dados coletados e processados durante o projeto ByBus. Ainda é possível notar na Figura 13 um recurso implementado no mapa, em que quando o marcador da parada é clicado, um *pop-up* aparece mostrando mais informações.

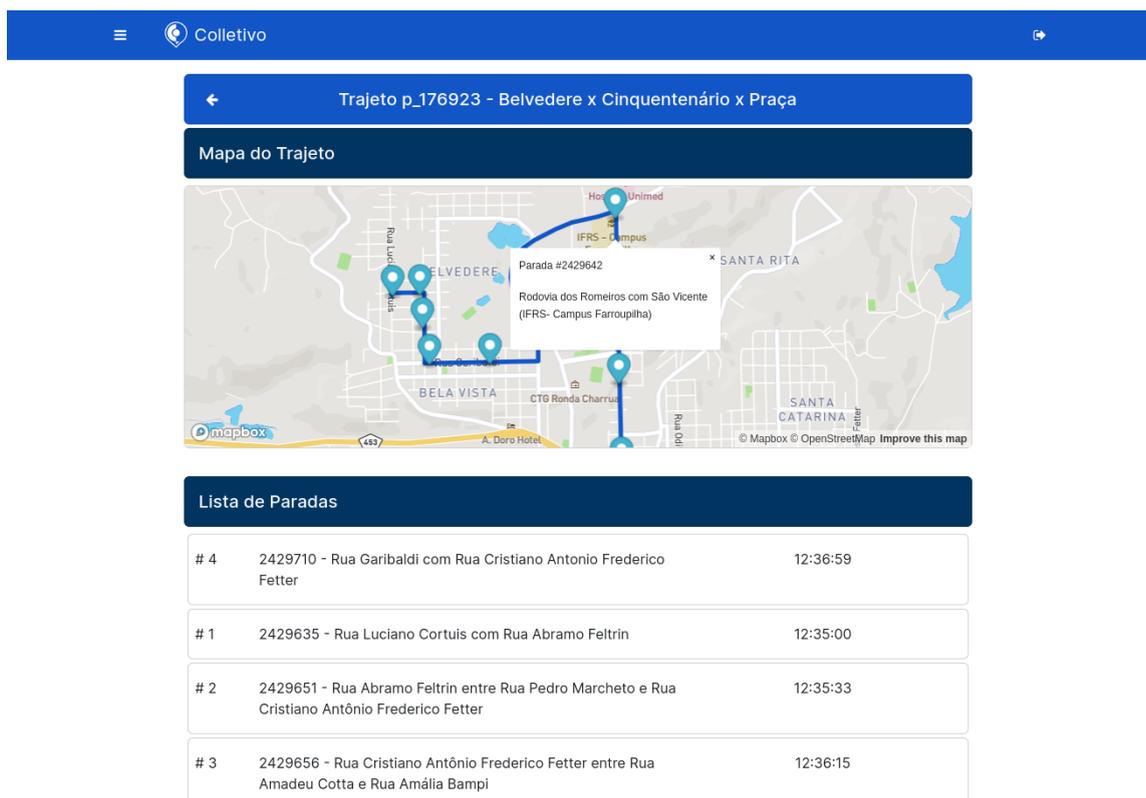


Figura 13. Interface mostrando dados de um trajeto real, coletado no projeto ByBus.

Fonte: Autoria própria.

5. Conclusão

Este trabalho apresentou o desenvolvimento do Colletivo, um sistema web que visa oferecer recursos para inserir, gerir, editar e exportar dados do formato GTFS. Por conta de algumas necessidades notadas ao longo da implementação do projeto ByBus, somado a demandas que surgiram após o fim do projeto, foi vista a oportunidade de criar uma plataforma gratuita e de código aberto para tratar as adversidades existentes.

Além das experiências como usuário de transporte público, foi de grande importância a participação como voluntário no projeto ByBus. Entender a realidade de motoristas, passageiros, especialistas em sistemas de mobilidade urbana, entre outros

envolvidos, foi de suma importância para poder desenvolver uma plataforma que facilite e motive a padronização dos dados. Por conta de toda essa complexidade, também é possível perceber o grande desafio que é entregar informações de transporte público com qualidade e precisão.

A aplicação oferece na API do *back-end* todas as ações necessárias para interagir com as entidades do GTFS, estruturadas na base de dados. Permitindo assim uma escalabilidade e flexibilidade, já que a API não se restringe apenas ao *front-end* desenvolvido, e pode ser acessada por outras aplicações. Assim, são contemplados os objetivos 01, 02 e 06.

Já a parte do *front-end* oferece uma interface amigável e simples para o usuário desenhar as rotas, ligando as paradas disponíveis e podendo ver o trajeto ser projetado no mapa. Esses dados são traduzidos para o formato GTFS, em que a base suporta, e assim podem ser visualizados em outro momento. Dessa forma é contemplado o objetivo 04 e parcialmente cumprido com o objetivo 07.

Contudo, os objetivos 03, 05 e 07 foram parcialmente alcançados. Não foi possível desenvolver alguns recursos desejados, como a importação e exportação dos dados por meio de arquivos, a edição e ajuste de visibilidade das rotas, e a adição e gerenciamento das paradas. Apesar disso, se entende que o que foi desenvolvido no momento servirá de base para esses novos recursos, já que o desenvolvimento e a arquitetura foram feitos considerando a escalabilidade da aplicação.

Para trabalhos futuros, além dos objetivos que não foram totalmente contemplados, pode-se projetar novos objetivos voltados ao GTFS Realtime (dados em tempo real). Esse tipo de suporte exige uma abordagem que se baseia no GTFS convencional porém possui suas particularidades. E por se tratar de um projeto que foi pensado em ser gratuito e de código aberto, novos recursos podem ser sugeridos e desenvolvidos pela comunidade, que terá acesso ao código por meio do GitHub.

Referências

AUTH0. jwt.io. JSON Web Tokens. [S.l.]. Auth0, 2023. Disponível em: <https://jwt.io/>. Acesso em: 12 jun. 2023.

BOOTSTRAP TEAM. Bootstrap. Build fast, responsive sites with Bootstrap. [S.l.]. Bootstrap, 2023. Disponível em: <https://getbootstrap.com/>. Acesso em: 8 jun. 2023.

BUTLER, Howard et al. The GeoJSON Format. In: *geojson.win*. [S.l.]. 2016. Disponível em: <https://stevage.github.io/geojson-spgeojson-spec.ec/>. Acesso em: 6 jun. 2023.

CARAVAN STUDIOS. Feito na Biblioteca: bybus. ByBus. 2023. Disponível em: <https://www.feitonabiblioteca.org/bybus?lang=pt>. Acesso em: 16 jun. 2023.

DBEAVER. DBeaver Community. Free Universal Database Tool. [S.l.]. DBeaver, 2023. Disponível em: <https://dbeaver.io/>. Acesso em: 11 jun. 2023.

FORTIN, Philippe; MORENCY, Catherine; TRÉPANIE, Martin. Innovative GTFS Data Application for Transit Network Analysis Using a Graph-Oriented Method. *Journal of Public Transportation*, Monteral, CA, ano 2016, n. 1077-291X, 1 dez. 2016. p. 18-37. Disponível em: https://www.researchgate.net/publication/310446593_Innovative_GTFS_Data_Application_for_Transit_Network_Analysis_Using_a_Graph-Oriented_Method. Acesso em: 28 mai. 2023.

GITHUB. GitHub: discover interesting projects and people to populate your personal news feed.. Discover interesting projects and people to populate your personal news feed.. 2023. Disponível em: <https://github.com/>. Acesso em: 24 mai. 2023.

GIT. Git: --distributed-even-if-your-workflow-isn't. 2023. Disponível em: <https://git-scm.com/>. Acesso em: 24 mai. 2023.

GOOGLE. Angular: one framework. mobile & desktop.. One framework. Mobile & desktop.. 2023. Disponível em: <https://angular.io/>. Acesso em: 16 mai. 2023.

IFRS - Campus Farroupilha. Aplicativo mobile a usuários de ônibus urbano chega a Farroupilha com projeto coletivo. 2018. Disponível em: <https://ifrs.edu.br/farroupilha/aplicativo-mobile-a-usuarios-de-onibus-urbano-chega-a-farroupilha-com-projeto-coletivo/>. Acesso em: 16 jun. 2020.

MAPBOX. Maps, geocoding, and navigation APIs & SDKs. Maps and location for developers Precise location data and powerful developer tools to change the way we navigate the world.. [S.l.]. Mapbox, 2023. Disponível em: <https://www.mapbox.com/>. Acesso em: 30 mai. 2023.

MCHUGH, Bibiana. Pioneering Open Data Standards: The GTFS Story. In: *Beyond Transparency. Beyond Transparency*. Portland, 2013. Disponível em: <https://beyondtransparency.org/chapters/part-2/pioneering-open-data-standards-the-gtfs-story/>. Acesso em: 18 set. 2022.

MICROSOFT. TypeScript. TypeScript is JavaScript with syntax for types. Microsoft, 2023. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 10 jun. 2023.

MICROSOFT. Visual Studio Code: code editing. redefined.. Code editing. Redefined.. 2023. Disponível em: <https://code.visualstudio.com/>. Acesso em: 11 jun. 2023.

MOBILITYDATA. General Transit Feed Specification. GTFS: Making Public Transit Data Universally Accessible. Montréal, QC, CA: MobilityData, 2023. Disponível em: <https://gtfs.org/>. Acesso em: 18 mai. 2023.

MOOVIT. Use a MaaS para simplificar a mobilidade. [S.l.]. Moovit Inc., 2023. Disponível em: <https://moovit.com/pt/>. Acesso em: 6 jun. 2023.

MYSLIWIEC, Kamil. NestJS. A progressive Node.js framework for building efficient, reliable and scalable server-side applications. NestJS, 2023. Disponível em:

<https://nestjs.com/>. Acesso em: 26 mai. 2023.

POSTGRESQL. PostgreSQL: the world's most advanced open source relational database. The World's Most Advanced Open Source Relational Database. 2023. Disponível em: <https://www.postgresql.org/>. Acesso em: 16 mai. 2023.

POSTMAN, INC.. Postman API Platform. [S.l.]. Postman, Inc., 2023. Disponível em: <https://www.postman.com/>. Acesso em: 11 jun. 2023.

RED HAT INC.. What is a REST API?. Understanding APIs. [S.l.]. Red Hat, Inc., 2020. Disponível em: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. Acesso em: 6 jun. 2023.

RODRIGUES, Marcos Antonio; SORRATINI, José Aparecido. A QUALIDADE NO TRANSPORTE COLETIVO URBANO. 2008. Curso de Engenharia Civil, Faculdade de Engenharia Civil, Universidade Federal de Uberlândia, Uberlândia, 2008. Disponível em: https://www.researchgate.net/profile/Jose-Sorratini/publication/267387600_A_QUALIDADE_NO_TRANSPORTE_COLETIVO_URBANO/links/549807360cf2c5a7e342874f/A-QUALIDADE-NO-TRANSPORTE-COLETIVO-URBANO.pdf. acesso em: 11 dez. 2022.

SCHNEIDER, ADOLFO HENRIQUE. Desenvolvimento web com Client Side Rendering: combinando Single Page Application e serviços de backend. Orientador: Marcelo Soares Pimenta. 2016. TCC (Graduação) - Curso de CIÊNCIA DA COMPUTAÇÃO, INSTITUTO DE INFORMÁTICA, UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, Porto Alegre, 2016. Disponível em: <https://www.lume.ufrgs.br/handle/10183/150910>. acesso em: 11 dez. 2022.

SMARTBEAR. Swagger. Bearer Authentication. [S.l.]. SmartBear Software, 2023. Disponível em: <https://swagger.io/docs/specification/authentication/bearer-authentication/>. Acesso em: 12 jun. 2023.

SOMMERVILLE, Ian. Software Engineering. 9. ed. Boston, MA, EUA: Pearson Education, Inc., 2011. 790 p.

TECHSOUP. Caravan Studios: we build apps that help communities organize, access, and apply local resources to their most pressing problems.. We build apps that help communities organize, access, and apply local resources to their most pressing problems.. 2023. Disponível em: <https://www.caravanstudios.org/>. Acesso em: 16 jun. 2023.

THE OPENJS FOUNDATION. Node.js. Node.js® is an open-source, cross-platform JavaScript runtime environment.. [S.l.]. The OpenJS Foundation, 2023. Disponível em: <https://nodejs.org/en/>. Acesso em: 12 jun. 2023.

TORRES, Alexandre et al. Twenty years of object-relational mapping: A survey on

patterns, solutions, and their implications on application design. Information and Software Technology: Elsevier, ed. 82, p. 1-18, fev. 2017. Mensal. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0950584916301859>. Acesso em: 6 jun. 2023.

TRANSIT. Transit: go your own way. Go Your Own Way. 2023. Disponível em: <https://transitapp.com/>. Acesso em: 17 mai. 2023.

TRILLIUM TRANSIT. Trillium - GTFS, Maps, Apps, and Websites for Public Transportation: we make transit easier to use. We make transit easier to use. 2023. Disponível em: <https://trilliumtransit.com/>. Acesso em: 30 mai. 2023.

TYPEORM. TypeORM: Beautiful and Powerful ORM for TypeScript and JavaScript (2023). Disponível em: <https://typeorm.io/>. Acesso em: 28 maio 2023.