

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JULIA TURAZZI ALMEIDA

BUS TRACKER: Um novo jeito de rastrear as linhas de ônibus!

RIO DE JANEIRO
2025

JULIA TURAZZI ALMEIDA

BUS TRACKER: Um novo jeito de rastrear as linhas de ônibus!

Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Orientador: Profa. Silvana Rossetto

RIO DE JANEIRO

2025

CIP - Catalogação na Publicação

A447b Almeida, Julia Turazzi
BUS TRACKER: um novo jeito de rastrear as linhas
de ônibus! / Julia Turazzi Almeida. -- Rio de
Janeiro, 2025.
98 f.

Orientadora: Silvana Rossetto.
Trabalho de conclusão de curso (graduação) -
Universidade Federal do Rio de Janeiro, Instituto
de Computação, Bacharel em Ciência da Computação,
2025.

1. Rastreamento de ônibus. 2. Mobilidade urbana.
3. Desenvolvimento web. 4. Engenharia de software.
I. Rossetto, Silvana, orient. II. Título.


JULIA TURAZZI ALMEIDA

BUS TRACKER: Um novo jeito de rastrear as linhas de ônibus!


Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 25 de Fevereiro de 2025


BANCA EXAMINADORA:

Documento assinado digitalmente
 **SILVANA ROSSETTO**
Data: 27/02/2025 16:29:52-0300
Verifique em <https://validar.iti.gov.br>

Silvana Rossetto, D.Sc.
Instituto de Computação - UFRJ

Documento assinado digitalmente
 **VIVIAN DOS SANTOS SILVA**
Data: 27/02/2025 17:56:00-0300
Verifique em <https://validar.iti.gov.br>

Vivian dos Santos Silva, Ph.D.
Instituto de Computação - UFRJ

Documento assinado digitalmente
 **LARISSA MONTEIRO DA FONSECA GALENO**
Data: 27/02/2025 19:18:20-0300
Verifique em <https://validar.iti.gov.br>

Larissa Monteiro da Fonseca Galeno
Instituto de Computação - UFRJ

Dedico este trabalho à família Buscapé.

AGRADECIMENTOS

Quero agradecer, antes de tudo, aos meus pais, que sempre me deram as oportunidades e o apoio necessários para que eu pudesse chegar até aqui. Aos meus irmãos, por estarem sempre por perto, compartilhando carinho, companhia e bons momentos ao longo dessa jornada. À Mel, por sua companhia inestimável e por sempre trazer leveza e diversão nos momentos de estresse.

Ao Arthur Valls, que esteve ao meu lado durante toda a graduação, compartilhando desafios, aprendizados e muitas risadas. Sua parceria foi essencial não só nessa trajetória acadêmica, mas em tantos outros projetos dentro e fora da faculdade – e, acima de tudo, nos meus momentos mais felizes.

Aos amigos da graduação, especialmente à Tropa de Choque, por enfrentarmos juntos cada obstáculo e tornarmos essa caminhada muito mais leve e inesquecível.

Ao Rafael Valls, pelas dicas valiosas e por compartilhar seu conhecimento sobre design, que fizeram toda a diferença no desenvolvimento deste e de outros trabalhos.

À minha orientadora, professora Silvana Rossetto, por aceitar esse desafio comigo, sempre com paciência e dedicação. Sua orientação foi essencial para que este trabalho chegasse a bom termo.

Também sou grata aos professores, à coordenação, à secretaria e a todos os profissionais da UFRJ, que contribuíram de alguma forma para minha formação.

Por fim, expresso minha gratidão a todos que, de maneira direta ou indireta, contribuíram para essa trajetória.

*“If you focus on what you left behind,
you will never see what lies ahead.”*

Gusteau, Ratatouille

*“It is not difficult to see yourself through the lens of other people,
and to wish you were all the different kaleidoscopic versions of you
they wanted you to be.*

*It is easy to regret, and keep regretting, ad infinitum, until our
time runs out.*

But it is not lives we regret not living that are the real problem.

*It is the regret itself. It’s the regret that makes us shrivel and
wither and feel like our own and other people’s worst enemy.*

*We can’t tell if any of those other versions would have been
better or worse.*

*Those lives are happening, it is true, but you are happening as well,
and that is the happening we have to focus on.”*

Matt Haig, The Midnight Library

RESUMO

A mobilidade urbana enfrenta desafios significativos, especialmente em relação à eficiência do transporte público. A falta de previsibilidade dos horários dos ônibus impacta diretamente a experiência dos usuários, levando à redução no número de passageiros nos últimos anos. Este trabalho apresenta o *Bus Tracker*, um sistema desenvolvido para oferecer informações em tempo real sobre a localização dos ônibus e alertas personalizados para usuários do transporte público. O objetivo principal é proporcionar uma experiência mais eficiente e confiável, incentivando o uso do transporte coletivo. O sistema foi desenvolvido utilizando uma arquitetura monolítica, caracterizada por uma única base de código que integra todos os componentes do sistema. Durante a fase de testes e avaliação com usuários, o sistema demonstrou eficiência na entrega das informações, recebendo feedback positivo quanto à sua usabilidade e impacto na rotina dos passageiros. Como contribuição, o *Bus Tracker* se diferencia por ser acessível via navegador sem necessidade de instalação, permitir personalização detalhada dos alertas e ser totalmente gratuito. Como trabalhos futuros, sugere-se a implementação de modelos preditivos para estimar tempos de chegada com base em padrões históricos de tráfego, bem como a integração com novos dados, como informações sobre congestionamentos e condições climáticas.

Palavras-chave: rastreamento de ônibus; mobilidade urbana; desenvolvimento web; engenharia de software.

ABSTRACT

Urban mobility faces significant challenges, especially regarding the efficiency of public transportation. The lack of predictability in bus schedules directly impacts user experience, leading to a decline in the number of passengers in recent years. This work presents *Bus Tracker*, a system developed to provide real-time information on bus locations and personalized alerts for public transport users. The main goal is to offer a more efficient and reliable experience, encouraging the use of public transportation. The system was developed using a monolithic architecture, characterized by a single codebase that integrates all system components. During the testing and user evaluation phase, the system demonstrated efficiency in delivering information, receiving positive feedback regarding its usability and impact on passengers' daily routines. As a contribution, *Bus Tracker* stands out for being accessible via a web browser without requiring installation, allowing detailed customization of alerts, and being entirely free. Future work includes the implementation of predictive models to estimate arrival times based on historical traffic patterns, as well as the integration of new data, such as congestion and weather conditions.

Keywords: bus tracking; urban mobility; web development; software engineering.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Diagrama de Casos de Uso - <i>Bus Tracker</i> | 24 |
| Figura 2 – Diagrama de Arquitetura do <i>Bus Tracker</i> | 35 |
| Figura 3 – Tela Inicial - usuário autenticado (primeira versão) | 39 |
| Figura 4 – Tela Inicial - usuário não autenticado (primeira versão) | 40 |
| Figura 5 – Informações e localização dos ônibus no mapa (primeira versão) | 41 |
| Figura 6 – Tela Inicial - usuário autenticado | 42 |
| Figura 7 – Tela Inicial - usuário não autenticado | 42 |
| Figura 8 – Tela Inicial - modo escuro | 43 |
| Figura 9 – Tela Inicial - barra lateral | 43 |
| Figura 10 – Esquema do Banco de Dados | 48 |
| Figura 11 – Tempo de Resposta HTTP sem <i>cache</i> vs Número de Usuários | 57 |
| Figura 12 – Tempo de Resposta HTTP com <i>cache</i> vs Número de Usuários | 58 |
| Figura 13 – Comparação da média do Tempo de Resposta HTTP com <i>cache</i> e sem <i>cache</i> por Número de Usuários | 58 |
| Figura 14 – Login | 82 |
| Figura 15 – Guia de Utilização do <i>Bus Tracker</i> | 82 |
| Figura 16 – Alerta Registrado | 83 |
| Figura 17 – Alertas Cadastrados | 83 |
| Figura 18 – Tabela das Informações dos Ônibus | 84 |
| Figura 19 – Email de Verificação | 84 |
| Figura 20 – Email de Atualizações | 85 |
| Figura 21 – Modo Mobile | 85 |

LISTA DE CÓDIGOS

| | | |
|----------|---|----|
| Código 1 | Exemplo de Resposta da API do Data Rio | 38 |
|----------|---|----|

LISTA DE QUADROS

| | |
|---|----|
| Quadro 1 – Especificação de casos de uso: <i>Login</i> | 25 |
| Quadro 2 – Especificação de casos de uso: Cadastrar novo alerta | 26 |
| Quadro 3 – Especificação de casos de uso: Gerenciar alertas cadastrados | 27 |
| Quadro 4 – Especificação de casos de uso: Apagar alerta | 28 |
| Quadro 5 – Especificação de casos de uso: Visualizar informações dos ônibus | 29 |
| Quadro 6 – Especificação de casos de uso: Verificar localização dos ônibus em um mapa | 30 |
| Quadro 7 – Especificação de casos de uso: Acessar instruções do sistema | 31 |
| Quadro 8 – Especificação de casos de uso: Realizar cadastro | 32 |
| Quadro 9 – Especificação de casos de uso: Alterar senha | 33 |
| Quadro 10 – Especificação de casos de uso: Visualizar informações do alerta cadastrado no <i>email</i> | 34 |
| Quadro 11 – <i>Endpoints</i> Implementados | 47 |
| Quadro 12 – Conformidade com os Requisitos Funcionais | 61 |
| Quadro 13 – Conformidade com os Requisitos Não Funcionais | 62 |
| Quadro 14 – Conformidade com as Restrições | 62 |
| Quadro 15 – Respostas sobre o que os usuários gostam no aplicativo/ferramenta que utilizam - Parte 1 | 77 |
| Quadro 16 – Respostas sobre o que os usuários gostam no aplicativo/ferramenta que utilizam - Parte 2 | 78 |
| Quadro 17 – Respostas sobre os recursos que faltam nos aplicativos/ferramentas utilizados - Parte 1 | 78 |
| Quadro 18 – Respostas sobre os recursos que faltam nos aplicativos/ferramentas utilizados - Parte 2 | 79 |
| Quadro 19 – Respostas sobre experiências e funcionalidades desejadas em aplicativos de transporte público - Parte 1 | 80 |
| Quadro 20 – Respostas sobre experiências e funcionalidades desejadas em aplicativos de transporte público - Parte 2 | 81 |
| Quadro 21 – Respostas sobre aspectos positivos do aplicativo | 93 |
| Quadro 22 – Respostas sobre aspectos frustrantes ou negativos do aplicativo | 93 |
| Quadro 23 – Respostas sobre funcionalidades desejadas no aplicativo | 94 |
| Quadro 24 – Respostas sobre erros ou problemas técnicos encontrados durante os testes | 96 |
| Quadro 25 – Respostas sobre comentários ou sugestões adicionais sobre o Bus Tracker | 98 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|---------------------------------------|
| API | Application Programming Interface |
| ASGI | Asynchronous Server Gateway Interface |
| DNS | Domain Name System |
| DOM | Document Object Model |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Tokens |
| YAML | Yet Another Markup Language |

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 14 |
| 1.1 | OBJETIVO | 15 |
| 1.2 | METODOLOGIA E RESULTADOS | 15 |
| 1.3 | ESTRUTURA DO TEXTO | 16 |
| 2 | TRABALHOS RELACIONADOS | 17 |
| 3 | ANÁLISE DE REQUISITOS | 20 |
| 3.1 | PESQUISA DE DEMANDA | 20 |
| 3.2 | LEVANTAMENTO DE REQUISITOS | 21 |
| 3.2.1 | Requisitos Funcionais | 21 |
| 3.2.2 | Requisitos Não Funcionais | 22 |
| 3.2.3 | Restrições | 23 |
| 3.3 | DIAGRAMA DE CASOS DE USO | 23 |
| 3.3.1 | Especificação dos Casos de Uso | 25 |
| 4 | ARQUITETURA E <i>DESIGN</i> DO SISTEMA | 35 |
| 4.1 | DIAGRAMA DA ARQUITETURA DO SISTEMA | 35 |
| 4.1.1 | <i>Backend</i> | 35 |
| 4.1.2 | <i>Frontend</i> | 37 |
| 4.1.3 | Infraestrutura | 38 |
| 4.2 | <i>DESIGN</i> DA INTERFACE DO USUÁRIO | 39 |
| 4.2.1 | Telas Seleccionadas | 39 |
| 4.2.2 | <i>Feedback</i> e Interações | 41 |
| 4.2.3 | Telas Atualizadas | 42 |
| 5 | IMPLEMENTAÇÃO | 44 |
| 5.1 | TRATAMENTO DE DADOS DOS PONTOS DE ÔNIBUS | 44 |
| 5.2 | DESENVOLVIMENTO DO <i>BACKEND</i> | 45 |
| 5.2.1 | Estrutura | 45 |
| 5.2.2 | Endpoints | 46 |
| 5.2.3 | Desafios e Soluções Implementadas | 46 |
| 5.3 | BANCO DE DADOS | 48 |
| 5.3.1 | Estrutura | 48 |
| 5.3.2 | Funcionalidades | 49 |
| 5.4 | DESENVOLVIMENTO DO <i>FRONTEND</i> | 50 |

| | | |
|----------|---|-----------|
| 5.4.1 | Estrutura | 50 |
| 5.4.2 | Funcionalidades | 50 |
| 5.4.3 | Abordagem de Desenvolvimento | 51 |
| 5.5 | <i>DEPLOY</i> | 52 |
| 5.6 | DA PROTOTIPAGEM AO CÓDIGO | 52 |
| 5.6.1 | Principais Desafios e Soluções | 53 |
| 5.7 | RESULTADO DAS TELAS DO SISTEMA | 53 |
| 6 | AVALIAÇÃO | 55 |
| 6.1 | ANÁLISE DE DESEMPENHO | 55 |
| 6.1.1 | Testes de Desempenho | 55 |
| 6.2 | COMPARAÇÃO COM OUTRO APLICATIVO | 58 |
| 6.3 | FEEDBACK DOS USUÁRIOS | 59 |
| 6.3.1 | Pontos de Melhoria e Problemas Técnicos | 60 |
| 6.3.2 | Funcionalidades Sugeridas e Comentários Finais | 60 |
| 6.4 | CONFORMIDADE COM OS REQUISITOS DEFINIDOS | 61 |
| 7 | CONCLUSÃO | 63 |
| 7.1 | TRABALHOS FUTUROS | 63 |
| | REFERÊNCIAS | 65 |
| | APÊNDICE A – FORMULÁRIO DE PESQUISA DE DEMANDA | 70 |
| | APÊNDICE B – RESULTADOS DO FORMULÁRIO DE PESQUISA DE DEMANDA | 74 |
| | APÊNDICE C – RESULTADOS DAS TELAS DO <i>BUS TRACKER</i> | 82 |
| | APÊNDICE D – FORMULÁRIO DE AVALIAÇÃO | 86 |
| | APÊNDICE E – RESULTADOS DO FORMULÁRIO DE AVALIAÇÃO | 91 |

1 INTRODUÇÃO

A mobilidade urbana no Brasil enfrenta desafios significativos, especialmente relacionados à eficiência e qualidade do transporte público. Nas últimas décadas, observou-se uma redução de mais de 30% na utilização do transporte coletivo no país (CNT, 2024). Essa crise está diretamente vinculada ao modelo de financiamento adotado. Enquanto países desenvolvidos implementaram soluções baseadas em sustentabilidade econômica, o Brasil ainda não avançou nesse sentido (MOBILIDADE, 2023).

Um panorama geral do transporte público coletivo nos municípios brasileiros revela dificuldades comuns, como a redução da oferta de serviços e o aumento dos custos operacionais, que sobrecarregam tanto o poder público quanto os usuários. Segundo um estudo técnico da Confederação Nacional de Municípios (CNM), muitos municípios enfrentam problemas financeiros para manter a operação de frotas de ônibus, devido à ausência de políticas públicas integradas e à baixa arrecadação de tarifas. O estudo reforça a necessidade de reestruturar os modelos de financiamento e implementar soluções que equilibrem sustentabilidade econômica, qualidade do serviço e acessibilidade social (CNM, 2024).

No município do Rio de Janeiro, o cenário reflete os desafios nacionais, agravados por questões locais. Estudos indicam que, em uma década, os transportes coletivos do município perderam cerca de um bilhão de usuários, refletindo a urgência de melhorias na qualidade e confiabilidade dos serviços oferecidos (ARAUJO; SCHMIDT, 2025). Entre 2019 e 2024, houve uma redução de 36,64% no número de linhas de ônibus em operação, passando de 715 para 453 linhas (GLOBO, 2025). Essa diminuição resultou em atrasos, superlotação e aumento do tempo de espera para os passageiros, desestimulando o uso do transporte coletivo e contribuindo para a preferência pelo transporte individual (GLOBO, 2025).

Embora existam aplicativos de rastreamento de ônibus, como **Moovit**, **CittaMobi**, **Cadê o Ônibus?**, **Quicko**, **Vá de Ônibus** e **Google Maps**, cada um apresenta limitações que podem impactar a experiência do usuário. Muitos exigem instalação no celular, consumindo armazenamento e restringindo o acesso para quem prefere soluções *web*. Além disso, funcionalidades avançadas, como notificações personalizáveis sobre a chegada dos ônibus, geralmente estão disponíveis apenas em versões pagas ou são limitadas.

Contudo, há espaço para soluções mais abrangentes que atendam às necessidades dos passageiros de maneira eficaz. A ausência de alertas por *email* configuráveis, que permitam aos usuários receber notificações com base em critérios específicos como horário e distância, reforça a necessidade de uma nova solução que ofereça um serviço mais acessível e flexível.

1.1 OBJETIVO

Este trabalho tem como objetivo principal desenvolver o aplicativo Bus Tracker, uma ferramenta destinada a aprimorar a experiência dos usuários de transporte público por ônibus. Embora o sistema tenha sido concebido como uma solução extensível para diferentes regiões, nesta versão, o escopo está restrito à cidade do Rio de Janeiro. Essa delimitação se deve à disponibilidade dos dados na API utilizada, bem como à necessidade de validar a solução dentro de um contexto específico antes de sua possível expansão.

A escolha pelo envio de alertas por *email* foi motivada pela facilidade de implementação e pelas limitações tecnológicas e financeiras relacionadas a outras APIs, como as do WhatsApp e SMS (BUSINESS, 2024) (TWILIO, 2024) (CLICKSEND, 2024). O uso dessas plataformas geralmente implica custos adicionais e restrições específicas, tornando o *email* uma alternativa mais viável para o projeto neste estágio inicial.

Os objetivos específicos incluem fornecer notificações em tempo real sobre a localização dos ônibus via *email*, permitindo que os usuários acompanhem os horários de chegada dos veículos nos pontos de escolhidos; oferecer um planejamento eficiente de horários com base nas preferências dos usuários; e contribuir para a melhoria da mobilidade urbana, incentivando o uso do transporte público por meio de informações precisas e acessíveis.

1.2 METODOLOGIA E RESULTADOS

A metodologia adotada teve como objetivo garantir um estudo abrangente e a completude da aplicação, desde a análise do problema até a validação do sistema. Para isso, foram consideradas diferentes etapas, incluindo pesquisa de mercado, definição da arquitetura, testes de escalabilidade e validação com usuários reais.

Inicialmente, foi realizada uma análise do problema por meio de pesquisas sobre a oferta de informações de transporte público em tempo real. Foram avaliadas soluções existentes e identificadas necessidades não atendidas, como a falta de notificações personalizadas sobre a chegada dos ônibus. Além disso, foram considerados dados da API do **Data Rio** para viabilizar um sistema que pudesse fornecer informações confiáveis aos usuários.

A arquitetura do sistema foi concebida para equilibrar eficiência e escalabilidade. Desenvolvido utilizando **Python** e **React**, o sistema foi estruturado para permitir tanto consultas em tempo real sobre a localização dos ônibus quanto a configuração de alertas personalizados, que são enviados por *email* aos usuários.

Para lidar com a alta demanda, foram conduzidos testes de escalabilidade, onde os testes simularam 1000 usuários, com um incremento de 100 usuários a cada 2 minutos, garantindo que o sistema suportasse acessos simultâneos sem degradação significativa de desempenho. A validação do sistema foi feita com usuários reais, focando na usabilidade e na efetividade dos alertas. Os principais resultados observados incluem: redução do

tempo de espera percebido e maior previsibilidade na chegada dos ônibus; interface intuitiva facilitando o uso por diferentes perfis de usuários; desempenho satisfatório, mesmo sob carga elevada, mantendo um tempo de resposta adequado para consultas e envio de notificações.

1.3 ESTRUTURA DO TEXTO

A estrutura desta monografia está organizada em capítulos, cada um abordando um aspecto específico do desenvolvimento do sistema *Bus Tracker*. No Capítulo 2 são analisados trabalhos relacionados, apresentando aplicativos existentes de rastreamento de ônibus, como **Moovit** e **CittaMobi**, destacando suas funcionalidades e limitações. No Capítulo 3 são demonstrados os resultados da pesquisa de demanda realizada com usuários, o Diagrama de Casos de Uso e a definição dos requisitos funcionais e não funcionais do sistema, considerando fatores como usabilidade, desempenho e segurança. No Capítulo 4 são descritos a estrutura técnica do *Bus Tracker*, incluindo sua arquitetura monolítica, os componentes do *backend* e *frontend*, além do *design* da interface do usuário. No Capítulo 5 são detalhados o desenvolvimento do *backend* e *frontend*, o tratamento dos dados dos pontos de ônibus, a estrutura do banco de dados e os desafios técnicos superados ao longo da construção do sistema. No Capítulo 6 são apresentados os testes de desempenho da API, a análise de gargalos do sistema e o *feedback* dos usuários sobre a experiência de uso, destacando sugestões de melhorias e funcionalidades desejadas. Por fim, no Capítulo 7 são avaliados os resultados do projeto em comparação com soluções existentes e propõem-se possíveis melhorias futuras, como a integração com mais APIs e o uso de inteligência artificial para modelos preditivos.

2 TRABALHOS RELACIONADOS

A pesquisa inicial na literatura científica buscou identificar estudos sobre o desenvolvimento e a implementação de sistemas de rastreamento de ônibus com notificações personalizáveis. No entanto, embora existam trabalhos que abordem o rastreamento de veículos, não foram encontrados artigos científicos que tratassem diretamente desse foco específico; a maioria dos estudos disponíveis discute, de forma genérica, as possíveis tecnologias aplicáveis à mobilidade urbana e aos sistemas de informação em tempo real no transporte público ou engenharia de software sem implementação (CEDER, 2021) (PAIVA et al., 2021) (WATKINS et al., 2011) (WONG; KOH; CHEONG, 2017) (ALVES et al., 2019) (ELIJAH et al., 2023). Diante desse cenário, optou-se por analisar aplicativos comerciais relevantes disponíveis no mercado. Os critérios para seleção incluíram a abrangência geográfica dos serviços, a diversidade e a relevância das funcionalidades oferecidas (como informações em tempo real e rastreamento de itinerários), bem como a popularidade e o impacto dos aplicativos no contexto urbano.

Atualmente, existem diversas aplicações no mercado que oferecem serviços de rastreamento de ônibus, cada uma com características e funcionalidades específicas que atendem a diferentes perfis de usuários. A seguir, são abordados alguns dos principais aplicativos disponíveis no mercado, incluindo o **Moovit**, **CittaMobi**, **Cadê o Ônibus?**, **Quicko**, **Vá de Ônibus** e **Google Maps**.

O **Moovit** destaca-se por sua abrangência, fornecendo informações em tempo real sobre horários e trajetos de ônibus, metrô, trens e outros meios de transporte em mais de 3.500 cidades ao redor do mundo, incluindo São Paulo e Rio de Janeiro. Sua interface intuitiva permite o planejamento de rotas, considerando estimativas de viagem e possíveis atrasos, além de notificações sobre alterações de horários e interrupções nos serviços. No entanto, algumas funcionalidades avançadas estão disponíveis apenas mediante assinatura paga, o que pode ser um obstáculo para alguns usuários (MOOVIT, 2024).

O **CittaMobi** diferencia-se pela precisão nas previsões de chegada dos ônibus nos pontos selecionados, ajudando os usuários a reduzir o tempo de espera. Atualmente, o aplicativo está disponível em mais de 300 cidades brasileiras, incluindo São Paulo, Salvador e Rio de Janeiro. Oferece funcionalidades como a possibilidade de adicionar à lista de favoritos linhas e paradas frequentes e permite a recarga de cartões de transporte em algumas localidades. Entretanto, a disponibilidade da recarga de cartões é limitada a regiões específicas, além de não disponibilizar notificações personalizáveis, o que pode ser uma limitação para usuários que buscam essas funcionalidades (CITTAMOBIL, 2024).

O aplicativo **Cadê o Ônibus?** é voltado principalmente para a cidade de São Paulo, oferecendo informações em tempo real sobre a localização dos ônibus. Com ele, os usuários podem acompanhar o deslocamento dos veículos e se programar adequadamente. O apli-

cativo também emite alertas sobre notícias relacionadas ao transporte público na capital paulista, como atualizações da SPTrans, do metrô e da CPTM. Contudo, sua limitação geográfica pode ser um impedimento para usuários de outras regiões. Além disso, o mesmo não oferece notificações personalizáveis, restringindo a capacidade dos usuários de receber alertas específicos conforme suas preferências (OLIVEIRA, 2024).

Já o **Quicko** era um aplicativo de mobilidade urbana que integrava diferentes modais de transporte, como ônibus, metrô, trens, bicicletas compartilhadas e serviços de transporte por aplicativo. Descontinuado em 2022, o aplicativo permitia aos usuários planejar rotas otimizadas com base no tempo e custo, oferecendo opções de trajetos mais rápidos ou mais econômicos. Fornecia informações em tempo real sobre o transporte público, incluindo notificações sobre alterações e intervenções nas linhas. Apesar de ter sido descontinuado, o **Quicko** é relevante para esta pesquisa devido à sua popularidade nas redes sociais e às funcionalidades que oferecia durante seu período de atividade. No entanto, mesmo durante seu funcionamento, o aplicativo não disponibilizava notificações personalizáveis, podendo limitar a experiência do usuário em termos de alertas específicos (QUICKO, 2024).

Exclusivamente para os usuários do transporte público do Rio de Janeiro, o **Vá de Ônibus** oferece informações abrangentes sobre as linhas de ônibus da região, incluindo consulta de itinerários, localização de pontos próximos e acompanhamento em tempo real. No entanto, sua funcionalidade é restrita a uma única cidade, além de não oferecer notificações personalizáveis, o que pode ser uma limitação para usuários que desejam alertas adaptados às suas necessidades (ÔNIBUS, 2024).

O **Google Maps** é uma ferramenta globalmente popular que fornece direções para motoristas e pedestres e disponibiliza informações sobre transporte público, como horários de partida e chegada, assim como opções de rotas integradas. Sua interface de fácil utilização e a integração com outros serviços do Google oferecem conveniência. Recentemente, o Google Maps incorporou funcionalidades baseadas em inteligência artificial, como a capacidade de encontrar lojas próximas que vendem produtos específicos e a "Vista em Vivo", que identifica estabelecimentos e estações próximas em tempo real. No entanto, a precisão das informações em tempo real pode variar dependendo da cidade e da atualização de dados pelas autoridades locais. Além disso, o Google Maps não oferece notificações personalizáveis para o transporte público, limitando a capacidade dos usuários de receber alertas específicos sobre suas rotas ou horários preferidos (GOOGLE, 2024b).

Embora essas aplicações ofereçam funcionalidades relevantes, ainda apresentam limitações que podem restringir o acesso pleno por parte dos usuários. Além disso, todas essas ferramentas exigem a instalação de aplicativos dedicados, consumindo espaço de armazenamento nos dispositivos móveis. Isso pode ser um problema para usuários com celulares de menor capacidade ou que preferem não instalar múltiplos aplicativos. A ausência de versões web acessíveis diretamente pelo navegador limita o acesso a esses serviços.

Outro ponto crítico é a falta de sistemas amplamente integrados de notificações personalizáveis. Recursos como alertas por *email* que permitam ao usuário definir janelas de horário específicas ou ajustar a distância do ponto escolhido para receber notificações sobre a aproximação do ônibus não são comumente encontrados gratuitamente.

Em contraste, a solução proposta visa superar essas limitações ao oferecer uma plataforma *web* acessível diretamente pelo navegador, eliminando a necessidade de instalação de aplicativos e economizando espaço nos dispositivos dos usuários. Além disso, planeja-se implementar um sistema de notificações personalizáveis, permitindo que os usuários recebam alertas por *email* com base em suas preferências de horário e distância. No entanto, funcionalidades como recarga de cartões de transporte e integração com outros modais, presentes em aplicativos como o **CittaMobi** e o **Quicko**, não estão previstas na solução proposta neste momento.

3 ANÁLISE DE REQUISITOS

O levantamento de requisitos foi conduzido por meio de pesquisa com potenciais usuários e análise de soluções já existentes. Com base nos dados obtidos, foram definidos os requisitos funcionais e não funcionais que orientam o desenvolvimento do sistema.

3.1 PESQUISA DE DEMANDA

Visando compreender as necessidades dos usuários, foi realizada uma pesquisa exploratória utilizando **Formulários do Google**. O questionário investigou hábitos, desafios e expectativas em relação ao uso de transporte público, com foco no ônibus.

O formulário, apresentado no Apêndice A, foi amplamente divulgado por redes sociais e compartilhado com indivíduos de diferentes faixas etárias e regiões do Rio de Janeiro.

As perguntas foram organizadas de forma dinâmica, adaptando-se às respostas dos participantes. Por exemplo, caso o usuário informasse que nunca utilizou aplicativos para transporte público, as perguntas relacionadas a esses aplicativos eram ocultadas automaticamente, garantindo uma experiência mais objetiva.

Durante o período de uma semana, foram coletadas 152 respostas, cujos dados estão disponíveis no Apêndice B. A análise permitiu identificar aspectos relevantes para o desenvolvimento do sistema, destacados a seguir:

- **Familiaridade Tecnológica:** A maioria dos participantes já utiliza aplicativos ou ferramentas relacionadas ao transporte público.
- **Cenário atual:** Foram relatadas dificuldades como incertezas nos horários dos ônibus, atrasos frequentes e a necessidade de utilizar múltiplos aplicativos para informações completas. Além disso, os participantes enfatizaram a ausência de alertas personalizados que comprometem o planejamento das viagens e a confiança no transporte público.
- **Expectativas:** Houve uma demanda por informações precisas sobre a localização e horários dos ônibus. Adicionalmente, os usuários destacaram a necessidade de interfaces intuitivas e notificações ajustáveis para facilitar o planejamento e aumentar a eficiência no dia a dia.

Dessa forma, com base nas respostas coletadas e na análise de ferramentas existentes, foram estabelecidos os seguintes objetivos para o aplicativo:

- Ser gratuita e acessível, disponível para usuários de diferentes faixas etárias e contextos socioeconômicos, eliminando a necessidade de instalação de aplicativos. Deve ser compatível com navegadores *web*, dispositivos **Android** e **iOS**.

- Prover informações precisas no tempo atual e oferecer atualizações confiáveis sobre a localização e horários dos ônibus.
- Oferecer uma experiência de uso simples e eficiente, por meio de uma interface de fácil utilização, adaptada tanto para iniciantes quanto para usuários experientes.
- Enviar alertas personalizados de acordo com as preferências individuais de horários e pontos de ônibus, otimizando o planejamento das viagens.
- Garantir a proteção de informações de autenticação do usuário por meio de criptografia.

3.2 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos do sistema foi conduzido com base na análise das necessidades identificadas na pesquisa exploratória com potenciais usuários e na avaliação de soluções similares já disponíveis. Os requisitos definidos foram organizados em duas categorias: Funcionais e Não Funcionais.

3.2.1 Requisitos Funcionais

O Requisito Funcional (RQF) especifica as funcionalidades que o sistema deve implementar para atender às necessidades dos usuários (ROMAN, 1985).

- **RQF1. Criar conta e realizar *login*:** Cadastrar usuários e permitir *login* seguro para acesso a funcionalidades avançadas.
- **RQF2. Visualizar localização dos ônibus em tempo atual:** Visualizar a localização em tempo atual dos ônibus por meio de um mapa interativo.
- **RQF3. Registrar alertas personalizados:** Registrar alertas personalizados, permitindo definir preferências de linha, ponto de ônibus, intervalo de tempo e distância.
- **RQF4. Receber notificações por email:** Receber notificações por email sempre que os critérios de alerta configurados forem atendidos.
- **RQF5. Gerenciar alertas cadastrados:** Gerenciar alertas cadastrados, permitindo visualização, edição e exclusão dos alertas registrados.
- **RQF6. Pesquisar pontos de ônibus:** Pesquisar pontos de ônibus por nome e visualizar informações no mapa interativo.
- **RQF7. Consultar guia de uso do sistema:** Consultar guia de uso do sistema, com explicações detalhadas sobre as funcionalidades, e acessar um email para esclarecer dúvidas.

3.2.2 Requisitos Não Funcionais

O Requisito Não Funcional (RQNF) define as características de qualidade e os padrões de desempenho esperados do sistema (PRESSMAN; MAXIM, 2021).

- **Desempenho**

- **RQNF1. Processamento rápido de consultas:** O sistema deve processar consultas e rastreamento de ônibus em menos de 1 segundo, garantindo uma experiência fluida (GOOGLE, 2017).
- **RQNF2. Desempenho sob alta carga:** O sistema deve evitar travamentos ou tempos de resposta altos das APIs, mesmo em situações de alta carga de dados ou acessos simultâneos.

- **Segurança**

- **RQNF3. Proteção de dados sensíveis:** O sistema deve garantir a segurança dos dados dos usuários por meio da criptografia das credenciais de *login*, por ser a única informação sensível manipulada.

- **Ambiente Operacional**

- **RQNF4. Compatibilidade com navegadores:** O sistema deve ser compatível com navegadores modernos (**Google Chrome**, **Mozilla Firefox**, **Microsoft Edge** e **Safari**) e dispositivos móveis (**Android** e **iOS**).

- **Confiabilidade**

- **RQNF5. Recuperabilidade:** O sistema deve ser capaz de retornar ao estado normal após falhas, com mecanismos de recuperação automática.

- **Usabilidade**

- **RQNF6. Interface adaptável:** O sistema deve fornecer uma interface responsiva e ajustável para dispositivos móveis e *desktops*.
- **RQNF7. Fácil utilização:** O sistema deve ser acessível para usuários com diferentes níveis de familiaridade tecnológica, priorizando operações simples e de fácil entendimento.

- **Disponibilidade e Manutenção**

- **RQNF8. Alta disponibilidade:** O sistema deve garantir que os usuários não sejam impactados por períodos de intermitência para manutenção, realizando-as fora dos horários de pico, quando os ônibus não estiverem em circulação.

3.2.3 Restrições

Durante o levantamento de requisitos, foram identificadas restrições que impactam o desenvolvimento e a operação do aplicativo. Essas restrições estabelecem limites técnicos, legais e operacionais que devem ser respeitados durante o projeto:

- **Conformidade Legal:** O aplicativo deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD) e outras regulamentações aplicáveis para garantir a proteção de dados pessoais dos usuários.
- **Acesso à Internet:** O aplicativo será acessível exclusivamente *online*, exigindo conexão à Internet para o funcionamento de suas principais funcionalidades, como rastreamento no tempo atual e cadastro de novas notificações.
- **Dados no Tempo Atual:** O aplicativo dependerá de APIs externas para obter dados no tempo atual sobre a localização dos ônibus. A qualidade e a frequência das atualizações estarão condicionadas às limitações dessas APIs.
- **Notificações por *Email*:** A escolha do *email* como meio de notificação foi determinada por restrições tecnológicas e financeiras. Alternativas como **WhatsApp** e **SMS** foram consideradas, mas apresentaram desafios como custos elevados, necessidade de integrações adicionais e restrições impostas pelas próprias APIs (BUSINESS, 2024) (TWILIO, 2024) (CLICKSEND, 2024). O *email*, por sua vez, oferece uma solução de fácil implementação e gratuita para o envio automatizado de alertas aos usuários.
- **Foco no Rio de Janeiro:** O aplicativo foi desenvolvido com foco no município do Rio de Janeiro devido à disponibilidade de dados públicos sobre a localização dos ônibus e à necessidade de atender aos prazos estabelecidos para a conclusão do projeto. A adaptação para outras cidades exigiria integração com novas fontes de dados e ajustes na infraestrutura, o que não foi viável dentro do tempo disponível.

3.3 DIAGRAMA DE CASOS DE USO

O Diagrama de Casos de Uso é uma ferramenta fundamental na Engenharia de Software, pois oferece uma representação visual das interações entre os atores (usuários ou

sistemas externos) e o sistema em desenvolvimento. Essa representação facilita a compreensão dos requisitos funcionais, permitindo que as partes interessadas identifiquem claramente as funcionalidades esperadas e as relações entre elas. Além disso, o Diagrama de Casos de Uso serve como uma base sólida para a comunicação entre analistas, desenvolvedores e clientes, assegurando que todos compartilhem uma visão comum do escopo do sistema. Ao ilustrar os casos de uso, o diagrama também auxilia na identificação de possíveis extensões ou inclusões de funcionalidades, promovendo uma modelagem mais estruturada e eficiente do *software* (LIU et al., 2003) (BARROS-JUSTO; BENITTI; TIWARI, 2019).

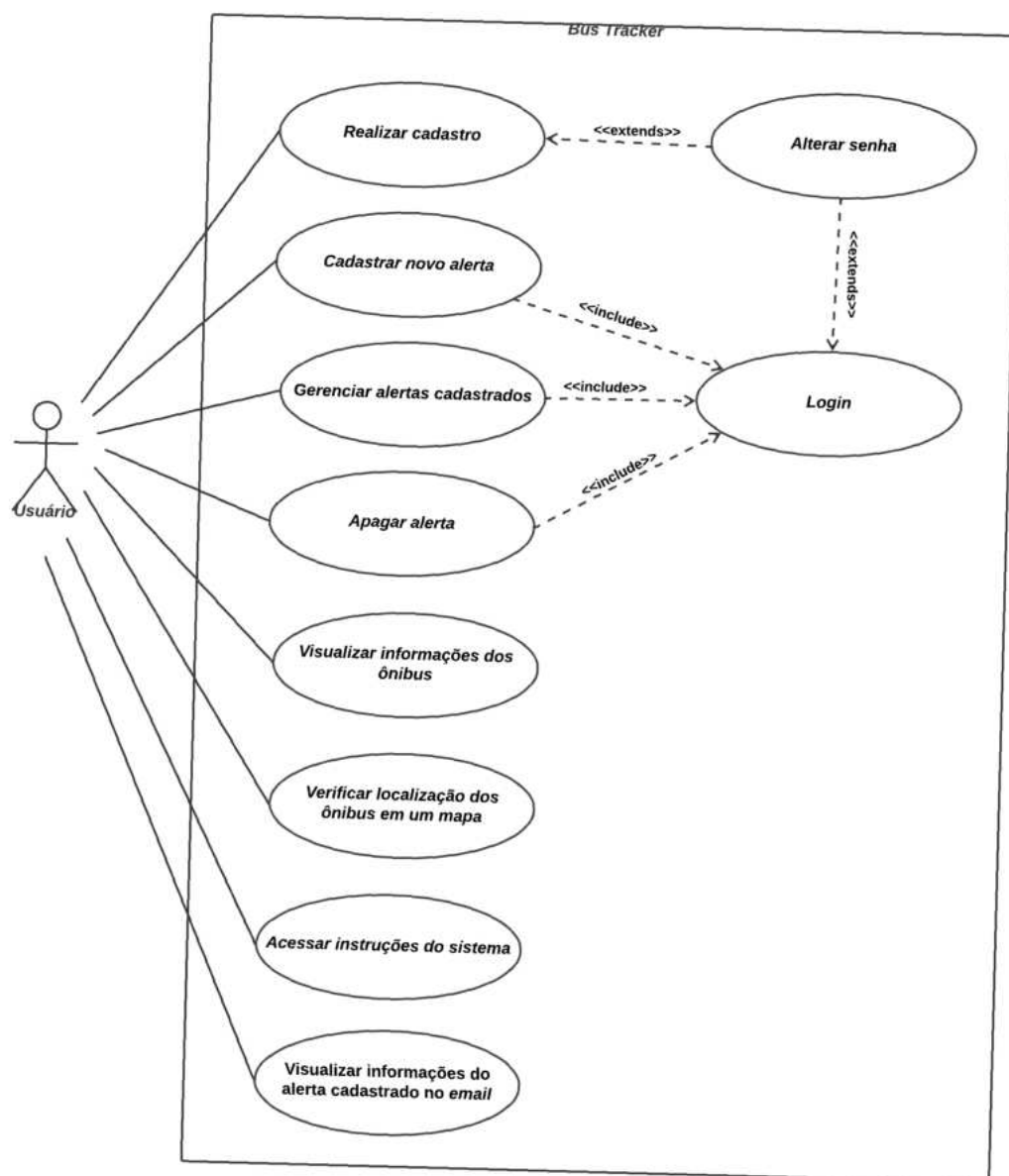


Figura 1 – Diagrama de Casos de Uso - *Bus Tracker*

3.3.1 Especificação dos Casos de Uso

A seguir, são detalhadas as especificações das principais funcionalidades do aplicativo.

| | |
|--------------------------------------|---|
| Caso de uso | <i>Login</i> |
| Descrição | Permite que o usuário se autentique no sistema para acessar funcionalidades avançadas. |
| Atores | Usuário |
| Pré-Condições | O usuário deve ter uma conta registrada no sistema. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário acessa o sistema na aba de <i>login</i>. • O sistema solicita as credenciais do usuário (<i>email</i> e <i>senha</i>). • O usuário insere as credenciais e envia. • O sistema valida as credenciais fornecidas. • O sistema autentica o usuário e o redireciona para a interface principal do sistema. <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • Credenciais inválidas: O sistema informa que as credenciais fornecidas são inválidas e solicita nova tentativa. • Recuperação de senha: O usuário acessa a opção de recuperação de senha caso não se lembre das credenciais. |
| Pontos de extensão | Pode incluir "Alterar senha" caso as credenciais estejam incorretas ou foram esquecidas. |
| Pós-condições | O usuário está autenticado no sistema e pode acessar as funcionalidades avançadas. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • <i>Email</i> • <i>Senha</i> |
| Observações | O sistema deve garantir segurança na autenticação e proteção dos dados do usuário. |

Quadro 1 – Especificação de casos de uso: *Login*

| | |
|--------------------------------------|---|
| Caso de uso | Cadastrar novo alerta |
| Descrição | Permite que o usuário configure um alerta para ser notificado sobre a aproximação de um ônibus, inserindo a linha, o ponto de ônibus, o intervalo de horário em que deseja receber a notificação (apenas no próprio dia), e a distância do ônibus (em minutos) até o ponto para que a notificação seja disparada dentro do intervalo. |
| Atores | Usuário |
| Pré-Condições | O usuário deve estar autenticado no sistema. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário, na tela principal, insere as informações como linha de ônibus, ponto de ônibus, intervalo de tempo e distância. • O usuário confirma selecionando o botão "Agendar". • O sistema valida os dados e registra o alerta. • O sistema informa ao usuário que o alerta foi cadastrado com sucesso. <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • Alerta já cadastrado: Caso o usuário já possua um alerta com as mesmas informações, o sistema retorna que esse alerta já foi cadastrado. |
| Pós-condições | O alerta é registrado e ativado no sistema. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • Linha de ônibus • Ponto de ônibus • Intervalo de tempo (Horário Inicial e Horário Final) • Distância do ônibus (em minutos) |

Quadro 2 – Especificação de casos de uso: Cadastrar novo alerta

| | |
|--------------------------------------|--|
| Caso de uso | Gerenciar alertas cadastrados |
| Descrição | Permite que o usuário visualize ou apague os alertas previamente configurados. |
| Atores | Usuário |
| Pré-Condições | O usuário deve estar autenticado no sistema. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário acessa a área "Todos os alertas" na barra lateral da tela principal. • O sistema exibe a lista de alertas configurados pelo usuário. <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • Lista vazia: O sistema informa que não há alertas configurados. |
| Pós-condições | Os alertas cadastrados são visualizados. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • Lista de alertas cadastrados com informações: <ul style="list-style-type: none"> – Linha de ônibus – Ponto de ônibus – Intervalo de tempo (Horário Inicial e Horário Final) – Distância do ônibus (em minutos) |

Quadro 3 – Especificação de casos de uso: Gerenciar alertas cadastrados

| | |
|--------------------------------------|--|
| Caso de uso | Apagar alerta |
| Descrição | Permite que o usuário exclua um alerta configurado para não receber mais notificações. |
| Atores | Usuário |
| Pré-Condições | O usuário deve estar autenticado no sistema e ter pelo menos um alerta configurado. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário acessa a área "Todos os alertas" na barra lateral da tela principal. • O sistema exibe a lista de alertas configurados. • O usuário seleciona o alerta que deseja apagar. • O sistema solicita confirmação para a exclusão. • O usuário confirma a exclusão. • O sistema remove o alerta e informa que a exclusão foi bem-sucedida. <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • Cancelamento da exclusão: O usuário cancela a operação e o alerta não é apagado. |
| Pós-condições | O alerta é removido do sistema. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • Lista de alertas cadastrados com informações: <ul style="list-style-type: none"> – Linha de ônibus – Ponto de ônibus – Intervalo de tempo (Horário Inicial e Horário Final) – Distância do ônibus (em minutos) |

Quadro 4 – Especificação de casos de uso: Apagar alerta

| | |
|--------------------------------------|---|
| Caso de uso | Visualizar informações dos ônibus |
| Descrição | Permite que o usuário veja informações atualizadas sobre os ônibus de uma linha específica em formato de tabela, incluindo a linha, o número do ônibus, velocidade e distância em minutos até o ponto selecionado. |
| Atores | Usuário |
| Pré-Condições | O usuário deve selecionar uma linha e um ponto de ônibus. O usuário autenticado deve selecionar a opção de "Usar horário atual". |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário, na tela principal, insere as informações da linha de ônibus. • O usuário confirma selecionando o botão "Enviar". • O sistema valida os dados. • O sistema exibe as informações sobre os ônibus da linha em formato de tabela, contendo: <ul style="list-style-type: none"> – Número do ônibus; – Velocidade atual; – Distância em minutos do ponto selecionado; – Linha do ônibus. |
| Pós-condições | O usuário obtém informações detalhadas e atualizadas sobre os ônibus da linha selecionada. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • Linha de ônibus selecionada • Número do ônibus • Velocidade atual • Distância em minutos do ponto selecionado |
| Observações | Quando o caso de uso "Cadastrar novo alerta" é executado, o caso de uso atual é acionado automaticamente. |

Quadro 5 – Especificação de casos de uso: Visualizar informações dos ônibus

| | |
|--------------------------------------|--|
| Caso de uso | Verificar localização dos ônibus em um mapa |
| Descrição | Permite que o usuário veja no tempo atual a localização dos ônibus de uma linha específica em um mapa interativo. Usuários não cadastrados podem selecionar a linha e o ponto de ônibus para visualizar as informações com base no horário atual. Usuários cadastrados podem fazer o mesmo ou configurar alertas que também exibem a localização no tempo atual. |
| Atores | Usuário |
| Pré-Condições | O usuário deve selecionar uma linha e ponto de ônibus. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário, na tela principal, insere as informações da linha de ônibus. • O usuário confirma selecionando o botão "Enviar". • O sistema valida os dados. • O sistema exibe o mapa com a localização atual dos ônibus da linha selecionada. <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • Nenhum ônibus disponível: O sistema informa que não há ônibus em operação para a linha no momento. • Linha ou ponto inválido: O sistema informa que os dados fornecidos não são válidos e solicita nova entrada. |
| Pós-condições | O usuário vê a localização atual dos ônibus no mapa. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • Ícone de um ônibus na latitude e longitude atual dos ônibus da linha selecionada em um mapa. |
| Observações | Quando o caso de uso "Cadastrar novo alerta" é executado, o caso de uso atual é acionado automaticamente. |

Quadro 6 – Especificação de casos de uso: Verificar localização dos ônibus em um mapa

| | |
|--------------------------------------|---|
| Caso de uso | Acessar instruções do sistema |
| Descrição | Permite que o usuário acesse o guia de uso com as instruções do sistema. Incluindo descrição das funcionalidades dos usuários cadastrados e não cadastrados |
| Atores | Usuário |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário acessa a aba "Instruções do <i>Bus Tracker</i>", presente na barra lateral da tela principal do sistema. • O sistema exibe as instruções ou o guia de uso e informações de contato para eventuais dúvidas. |
| Pós-condições | O usuário acessa as informações necessárias para operar o sistema. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • Guia de uso • Descrição das funcionalidades • Informações de contato para retirar dúvidas |

Quadro 7 – Especificação de casos de uso: Acessar instruções do sistema

| | |
|--------------------------------------|--|
| Caso de uso | Realizar cadastro |
| Descrição | Permite que o usuário crie uma conta no sistema, fornecendo informações essenciais para acessar funcionalidades avançadas. |
| Atores | Usuário |
| Pré-Condições | O usuário não deve possuir cadastro e não deve estar autenticado no sistema. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário acessa a aba de cadastro. • O sistema solicita as informações necessárias (<i>email</i>, senha). • O usuário preenche as informações e envia o formulário. • O sistema valida os dados enviando um <i>email</i> de confirmação para o <i>email</i> cadastrado. • Após a confirmação, o sistema cria a conta e confirma o sucesso do cadastro ao usuário. <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • Dados inválidos: O sistema informa erros, como <i>email</i> já registrado, e solicita correção. • <i>Email</i> não verificado: O sistema não permite o <i>login</i> se o <i>email</i> não for confirmado. |
| Pontos de extensão | Alterar senha - o usuário, após o cadastro, pode optar por mudar a senha cadastrada no sistema. |
| Pós-condições | A conta do usuário é criada no sistema. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • <i>Email</i> • Senha |

Quadro 8 – Especificação de casos de uso: Realizar cadastro

| | |
|--------------------------------------|---|
| Caso de uso | Alterar senha |
| Descrição | O sistema permite que o usuário atualize sua senha após o cadastro ou durante o <i>login</i> . |
| Atores | Usuário |
| Pré-Condições | O usuário deve possuir cadastro no sistema. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O usuário acessa a área "Esqueceu sua senha?". • O sistema solicita o <i>email</i> cadastrado. • O usuário insere a informação solicitada. • O sistema valida se o <i>email</i> está cadastrado e envia um <i>email</i> para redefinição de senha. • O usuário cria uma nova senha. • O sistema informa que a senha foi alterada com sucesso <p>Fluxos Alternativos:</p> <ul style="list-style-type: none"> • <i>Email</i> não cadastrado: O sistema informa que o <i>email</i> não está cadastrado. |
| Pós-condições | A senha do usuário é atualizada no sistema. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • <i>email</i> • Nova senha |

Quadro 9 – Especificação de casos de uso: Alterar senha

| | |
|--------------------------------------|--|
| Caso de uso | Visualizar informações do alerta cadastrado no <i>email</i> |
| Descrição | O sistema permite que o usuário visualize os detalhes do alerta cadastrado diretamente no <i>email</i> enviado pelo sistema. |
| Atores | Usuário |
| Pré-Condições | O usuário deve ter um alerta cadastrado no sistema. O horário atual deve estar dentro da janela de horário definida pelo usuário. |
| Fluxo de eventos | <p>Fluxo Básico:</p> <ul style="list-style-type: none"> • O sistema verifica se o horário atual está dentro da janela estipulada pelo usuário. • O sistema detecta os ônibus, correspondentes à linha escolhida, que aproximam-se do ponto de ônibus, definido no alerta cadastrado. • O sistema gera um <i>email</i> contendo as informações do ponto de ônibus, linha, número e distância em minutos (informações cadastradas no alerta) e o envia ao usuário. • O usuário acessa sua caixa de entrada e visualiza o <i>email</i>. <p>Fluxo Alternativo:</p> <ul style="list-style-type: none"> • O horário atual está fora da janela estipulada: O sistema não dispara o alerta. |
| Pós-condições | O usuário recebe o <i>email</i> com as informações do alerta, respeitando a janela de horário configurada. |
| Mini estrutura de informações | <ul style="list-style-type: none"> • <i>email do usuário</i> • Linha do ônibus • Ponto de ônibus • Distância (em minutos) • Intervalo de horário configurado |

Quadro 10 – Especificação de casos de uso: Visualizar informações do alerta cadastrado no *email*

4 ARQUITETURA E *DESIGN* DO SISTEMA

O *Bus Tracker* foi desenvolvido com uma arquitetura monolítica, caracterizada por uma única base de código que integra todos os componentes do sistema. Essa abordagem oferece simplicidade no desenvolvimento e na implantação, facilitando a manutenção do código. Além disso, a comunicação interna entre os componentes é mais eficiente, devido à ausência de sobrecarga de comunicação entre serviços independentes. No entanto, é importante notar que, à medida que a aplicação cresce, a escalabilidade e a flexibilidade podem se tornar desafiadoras, exigindo um planejamento cuidadoso para futuras expansões (HARRIS, 2023).

A seleção das tecnologias foi fundamentada em critérios como desempenho, facilidade de uso, suporte por uma comunidade ativa, compatibilidade com os requisitos do sistema e, sobretudo, na experiência prévia com a maioria das ferramentas escolhidas.

4.1 DIAGRAMA DA ARQUITETURA DO SISTEMA

O diagrama abaixo apresenta uma visão detalhada dos componentes do sistema, destacando as interações entre o *backend*, *frontend*, banco de dados, infraestrutura e APIs externas:



Figura 2 – Diagrama de Arquitetura do *Bus Tracker*

4.1.1 *Backend*

O *backend* do sistema foi desenvolvido utilizando as seguintes tecnologias, selecionadas por suas características específicas:

- **Linguagem de Programação: Python**, reconhecida por sua simplicidade e legibilidade, facilitando o desenvolvimento rápido. Sua vasta biblioteca padrão e a integração com diversos *frameworks* modernos, ideal para aplicações *web* complexas (RÍOS et al., 2016) (ROSSUM; DRAKE, 2009).
- **Framework Principal: FastAPI**, um *framework* robusto para a construção de APIs com **Python 3.6+** baseado em padrões como **OpenAPI**. Destaca-se por permitir a criação rápida de APIs de alto desempenho, além de oferecer suporte nativo para operações assíncronas (RAMÍREZ, 2024).
- **Servidor de Aplicação: Uvicorn**, um servidor ASGI (*Asynchronous Server Gateway Interface*), construído sobre o *loop* de eventos **uvloop** e **httptools**. É otimizado para servir aplicações assíncronas em **Python** (CHRISTIE, 2024).
- **Banco de Dados: MySQL**, um sistema de gerenciamento de banco de dados relacional amplamente utilizado que oferece suporte para operações transacionais e consultas (MYSQL, 2001).
- **Autenticação e Criptografia: python-jose**, biblioteca que facilita a codificação e decodificação de *JSON Web Tokens* (JWT), permitindo a implementação de sistemas de autenticação seguros (DAVIS, 2024). Além disso, o **bcrypt**, biblioteca utilizada para a criptografia de senhas, aplicando um algoritmo de *hashing* que aumenta a segurança contra ataques de força bruta (PYCA, 2024).
- **Notificações por Email: smtplib** e **email-validator**. A biblioteca **smtplib** é parte da biblioteca padrão do **Python** e fornece funções para envio de *emails* utilizando o protocolo SMTP (MONSYNE et al., 2024). Já o **email-validator** é uma biblioteca utilizada para validação de endereços de email, assegurando que os *emails* fornecidos estejam no formato correto e sejam entregáveis (TAUBERER, 2024).
- **Geolocalização e Roteamento: OpenRouteService**, uma API que fornece serviços de roteamento, geocodificação e outros recursos de geolocalização. É utilizada para calcular rotas otimizadas e fornecer direções precisas, melhorando a experiência do usuário no acompanhamento de trajetos. A escolha dessa API foi influenciada por sua política generosa de uso gratuito. No plano padrão, a API permite até 2.000 requisições diárias para direções (HEIGIT, 2024). Esses limites superam os de muitas outras APIs de geolocalização, como **Google Maps API** (GOOGLE, 2024a).
- **Manipulação de Dados: Pandas**, biblioteca desenvolvida para análise de dados em **Python**. Foi utilizada para processar o conjunto de dados de pontos de ônibus

do **Data Rio** (PREFEITURA-RJ, 2024b) em uma estrutura de dados de fácil utilização, permitindo a manipulação de atributos e a criação de um CSV atualizado (PANDAS, 2024).

- **Testes: Pytest**, uma estrutura de testes que facilita a escrita de casos de teste simples e escaláveis. Suporta uma variedade de recursos, incluindo plugins, permitindo a criação de testes para garantir a qualidade e a confiabilidade do código (KREKEL et al., 2004).

4.1.2 *Frontend*

O *frontend* foi desenvolvido com as seguintes ferramentas, cada uma escolhida por suas vantagens específicas:

- **Linguagem de Programação: JavaScript/TypeScript**. O **TypeScript**, uma extensão do **JavaScript**, adiciona tipagem estática ao código, permitindo a detecção de erros em tempo de desenvolvimento e facilitando a manutenção e escalabilidade da aplicação (BIERMAN; ABADI; TORGERSEN, 2014).
- **Framework Principal: React**, uma biblioteca para construção de interfaces de usuário baseada em componentes reutilizáveis. Facilita o desenvolvimento de interfaces interativas e dinâmicas, permitindo a atualização do DOM (*Document Object Model*) (META, 2024).
- **Gerenciamento de Estado: Redux Toolkit**, ferramenta oficial para configuração e gerenciamento do estado global em aplicações **React**. Simplifica a implementação de padrões complexos de gerenciamento de estado, reduzindo a quantidade de código *boilerplate* (trechos de código repetitivos e padronizados) (REDUX, 2024).
- **Estilização: TailwindCSS e Material-UI**. **TailwindCSS**, *framework* de utilitários CSS que permite a criação rápida de *designs* personalizados sem sair do HTML (TAILWIND, 2024). Já o **Material-UI** fornece uma coleção de componentes **React** prontos para uso, seguindo as diretrizes de *design* do **Material Design**, acelerando o desenvolvimento de interfaces (MATERIAL-UI, 2024).
- **Componentes e Utilidades: React Hook Form e ShadCN**. O **React Hook Form**, utilizado para facilitar o gerenciamento de formulários em React (REACT-HOOK, 2024). O **ShadCN** fornece uma coleção de componentes estilizados e acessíveis pré-prontos, acelerando o desenvolvimento de interfaces de usuário (SHADCN, 2024).
- **Mapas e Geolocalização: OpenLayers.js**, uma biblioteca **JavaScript** de código aberto para exibição de mapas interativos. Suporta uma variedade de fontes de

dados geoespaciais e formatos, permitindo a integração de mapas personalizados e funcionalidades avançadas de geolocalização na aplicação (OPENLAYERS, 2024).

- **Outras Dependências:** **Zod**, uma biblioteca de validação de esquemas **TypeScript-first**, permitindo a definição e verificação de formatos de dados de forma declarativa (MCDONNELL, 2024).

4.1.3 Infraestrutura

A infraestrutura do sistema foi planejada para garantir consistência, desempenho e facilidade de implantação:

- **Containerização:** **Docker** e **Docker Compose**. O **Docker** permite empacotar a aplicação e suas dependências em *container*, assegurando que o *software* funcione de maneira uniforme em diferentes ambientes. O **Docker Compose** facilita a definição e execução de aplicações *multicontainer*, permitindo gerenciar serviços, redes e volumes através de um único arquivo YAML (DOCKER, 2024b).
- **APIs:** As APIs, ou Interfaces de Programação de Aplicações, são conjuntos de padrões e protocolos que permitem a comunicação entre diferentes sistemas e aplicações. Elas funcionam como pontes que facilitam a integração de funcionalidades de um *software* em outro (GOODWIN, 2024).
 - **Transporte Público:** Integração com a API do **Data Rio**, que se propõe a ser uma fonte confiável de informações sobre ônibus e localização em tempo real (PREFEITURA-RJ, 2024a).

Exemplo de resposta da API:

Código 1 – Exemplo de Resposta da API do **Data Rio**

```
{
  "ordem": "A27625",
  "latitude": "-22.9728",
  "longitude": "-43.22477",
  "datahora": "1737044028000",
  "velocidade": "3",
  "linha": "LECD96",
  "datahoraenvio": "1737044031000",
  "datahoraservidor": "1737044037000"
}
```

- **Geolocalização:** **OpenRouteService**, utilizado para planejamento eficiente de rotas e cálculo de distâncias, essencial para o rastreamento em tempo real. A API do **OpenRouteService** fornece diversos endpoints para serviços de

roteamento, incluindo direções, isócronas (O'SULLIVAN; MORRISON; SHEARER, 2000) (uma área no mapa que mostra até onde você pode chegar em um certo tempo ou distância) e matrizes de distância (HEIGIT, 2024).

4.2 DESIGN DA INTERFACE DO USUÁRIO

A ferramenta **Figma** foi utilizada para a criação e interação de protótipos, permitindo testar diferentes abordagens de *design* com os usuários e para aprimorar rapidamente a interface. Escolhida pela facilidade de uso e compartilhamento de protótipos interativos, auxiliando nos testes com usuários e, também, pela experiência prévia própria utilizando a plataforma (FIGMA, 2024b).

4.2.1 Telas Selecionadas

Exemplos de telas criadas incluem:

- **Tela Inicial (usuários autenticados):** Exibe as opções de agendamento, com campos para preenchimento de "Linha de ônibus", "Ponto de ônibus", "Horário Inicial", "Horário Final" e "Distância do ônibus (em minutos)". O botão de "Agendar" permanece desativado até que todos os campos sejam preenchidos.

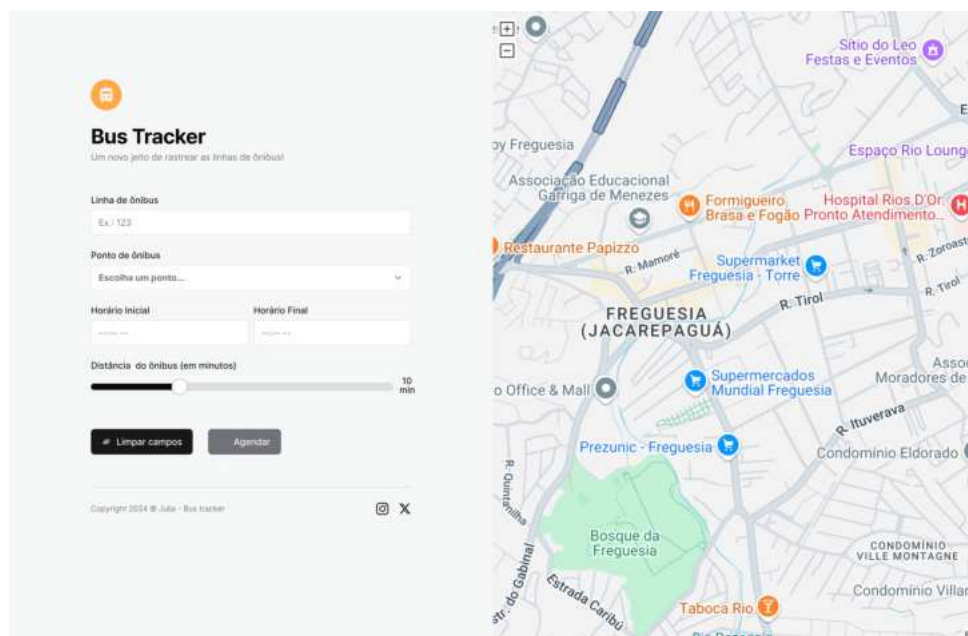


Figura 3 – Tela Inicial - usuário autenticado (primeira versão)

- **Tela Inicial (usuário não autenticado):** Exibe as opções de busca de ônibus, com campos para preenchimento de "Linha de ônibus" e "Ponto de ônibus". O botão de envio permanece desativado até que todos os campos sejam preenchidos.

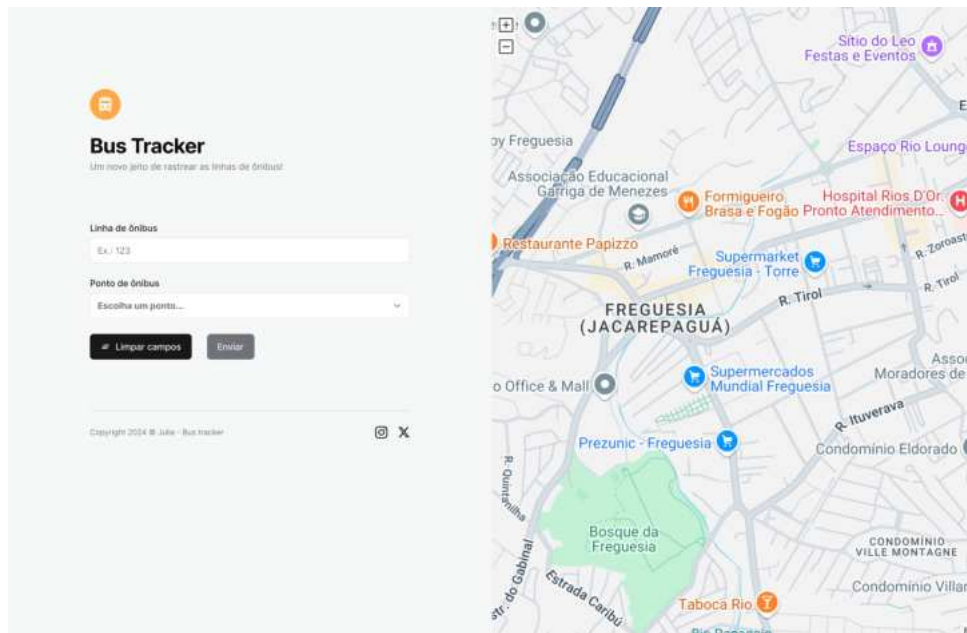


Figura 4 – Tela Inicial - usuário não autenticado (primeira versão)

- **Tela de Rastreamento e informação dos Ônibus:** Exibe a localização atual dos ônibus no mapa e informações como "Ordem" (número do ônibus), "Velocidade" e "Distância" (minutos).
Observação: Usuários logados também têm acesso a essa tela, ao selecionar a opção "Agendar" o alerta.

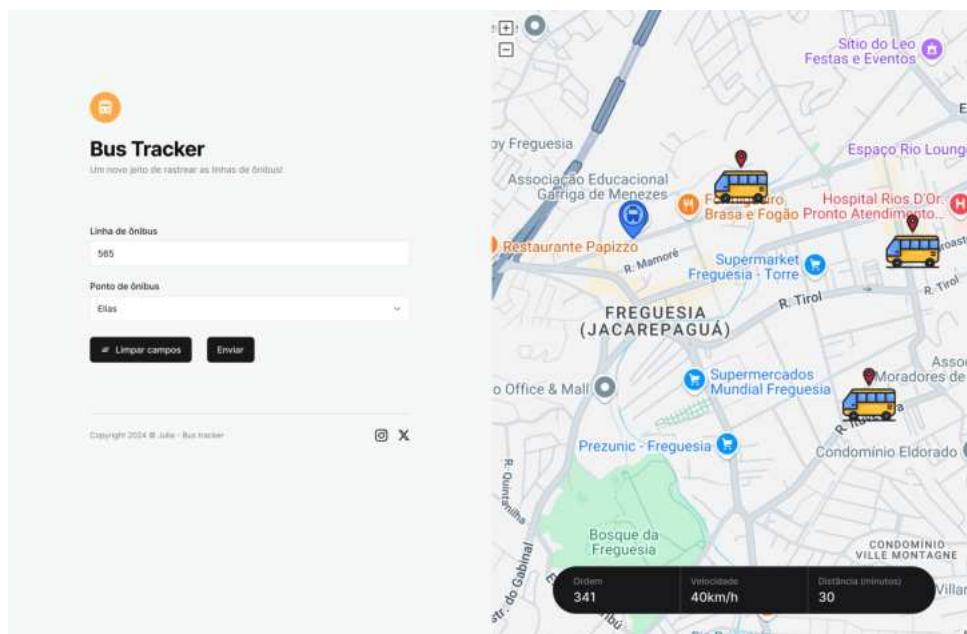


Figura 5 – Informações e localização dos ônibus no mapa (primeira versão)

4.2.2 *Feedback* e Interações

O *feedback* dos usuários foi essencial para moldar o *design* final. A avaliação foi realizada a partir de entrevistas com voluntários, incluindo desenvolvedores e *designers* com experiência em aplicações *web* voltadas para diferentes públicos-alvo, juntamente com o envio do *link* do protótipo do Figma. Essa abordagem permitiu coletar informações mais aprofundadas sobre a usabilidade e a experiência de uso, levando a alterações significativas no projeto.

Entre as principais mudanças, destacam-se a implementação de um modo escuro, atendendo à demanda por maior acessibilidade e conforto visual. Além disso, um novo botão "Usar tempo atual" foi introduzido, permitindo que usuários autenticados acessem rapidamente a funcionalidade de visualização de ônibus em tempo real sem a necessidade de se desconectar da conta, evitando o agendamento de alertas.

Outra melhoria de destaque foi a inclusão de uma barra lateral com uma aba dedicada ao gerenciamento de alertas, um recurso inexistente anteriormente, que trouxe mais organização e facilidade no gerenciamento de notificações. A barra lateral também incorporou um guia de uso, auxiliando novos usuários na resolução de dúvidas e tornando o sistema mais intuitivo.

Por fim, foi adicionada uma funcionalidade de recuperação de senha, permitindo que os usuários redefinam suas credenciais de forma prática e segura, garantindo uma experiência mais confiável e acessível.

4.2.3 Telas Atualizadas

- Tela Inicial (usuário autenticado):

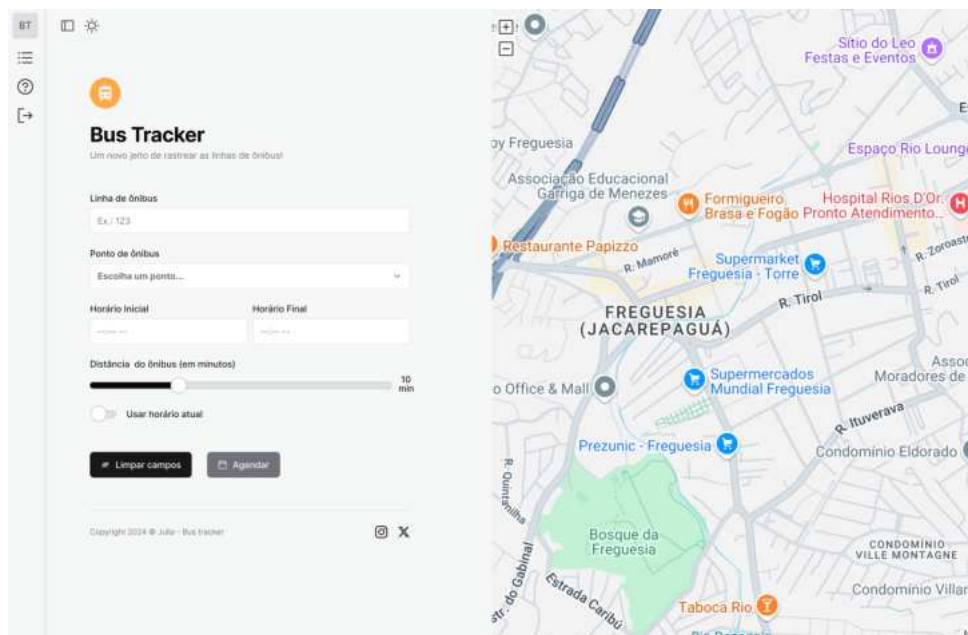


Figura 6 – Tela Inicial - usuário autenticado

- Tela Inicial (usuário não autenticado):

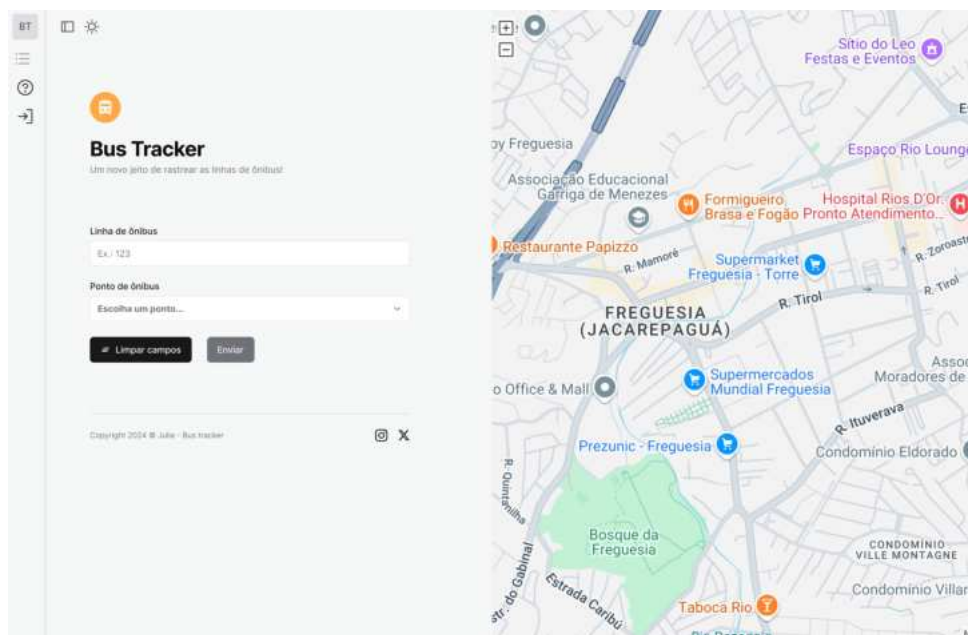


Figura 7 – Tela Inicial - usuário não autenticado

- Modo escuro:

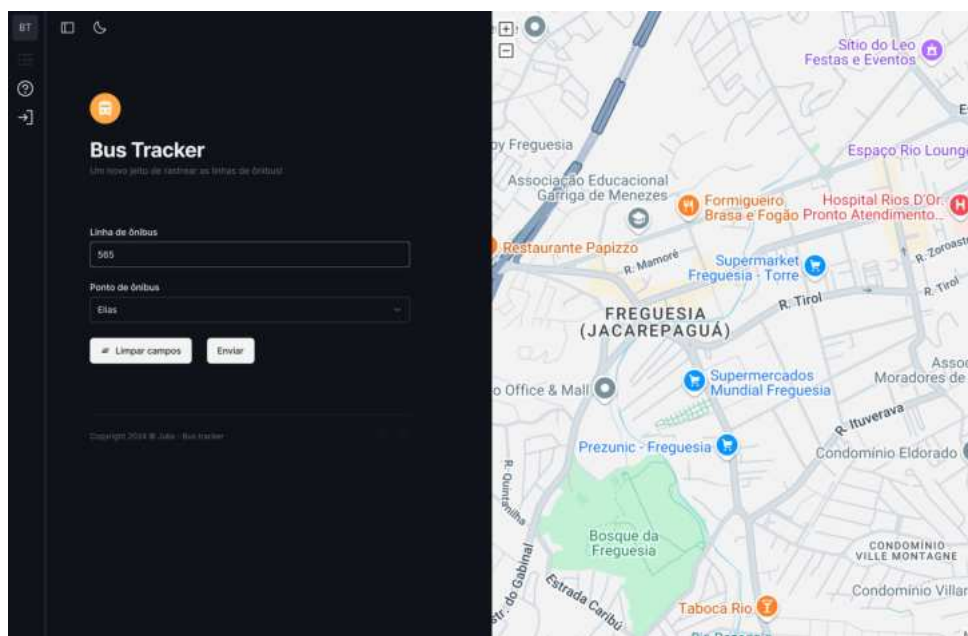


Figura 8 – Tela Inicial - modo escuro

- Barra Lateral:

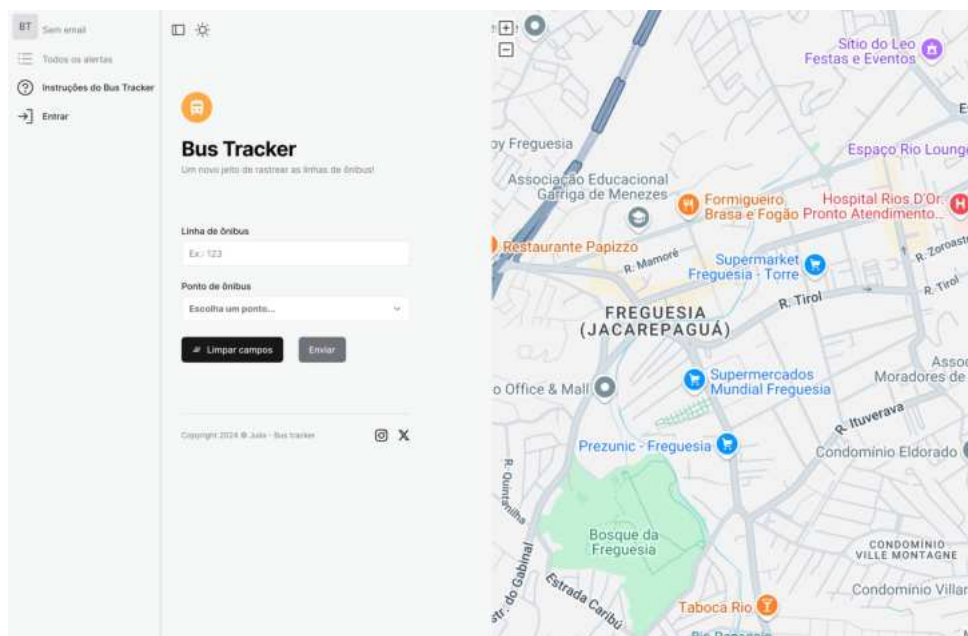


Figura 9 – Tela Inicial - barra lateral

5 IMPLEMENTAÇÃO

Este capítulo apresenta a estrutura e os principais aspectos técnicos do *Bus Tracker*, desde o tratamento de dados até o desenvolvimento do *backend* e *frontend*. Serão descritas as abordagens utilizadas para processar as informações dos pontos de ônibus, a organização do código, os *endpoints* da API e os desafios enfrentados. Também será detalhado o processo de *deploy*, incluindo as ferramentas escolhidas para hospedagem e segurança. Por fim, será discutida a transição do *design* para o código, utilizando o protótipo original como base.

Todo o código do projeto está disponível publicamente no GitHub através do *link*: <https://github.com/juliaturazzi/bus-tracker>. O mesmo está aberto para sugestões, reportes de erros ou propostas de melhorias.

5.1 TRATAMENTO DE DADOS DOS PONTOS DE ÔNIBUS

Os dados dos pontos de ônibus foram obtidos do portal oficial do Data Rio, uma iniciativa da Prefeitura do Rio de Janeiro que disponibiliza informações atualizadas sobre diversos aspectos da cidade, incluindo a rede de transporte público (PREFEITURA-RJ, 2024b). Esses dados foram selecionados por serem oficiais e abrangentes, auxiliando na confiabilidade.

O conjunto de dados original, disponível no formato CSV, contém as seguintes colunas: `stop_id`, `stop_code`, `stop_name`, `stop_desc`, `stop_lat`, `stop_lon`, `zone_id`, `stop_url`, `location_type`, `parent_station`, `stop_timezone`, `wheelchair_boarding` e `platform_code` (PREFEITURA-RJ, 2024b). No entanto, para a análise proposta neste trabalho, apenas as informações relacionadas ao nome do ponto de ônibus (`stop_name`), latitude (`stop_lat`) e longitude (`stop_lon`) foram utilizadas. O processo de tratamento de dados foi realizado em **Python** (ROSSUM; DRAKE, 2009) utilizando a biblioteca **Pandas**, que contém módulos preparados para manipulação de arquivos CSVs (PANDAS, 2024), para o tratamento do arquivo dos pontos de ônibus.

Durante o tratamento, identificou-se que alguns pontos de ônibus possuíam nomes duplicados, mas coordenadas diferentes. Isso ocorria porque esses pontos utilizavam o nome da rua como identificação, resultando em múltiplas paradas distintas com o mesmo nome ao longo de uma via. Além disso, esses pontos não tinham nenhum identificador adicional que permitisse diferenciá-los diretamente.

Para resolver essa questão e evitar ambiguidade, foi implementado um processo de renomeação. O primeiro ponto de ônibus com determinado nome foi mantido com o nome original, enquanto os demais receberam um índice numérico sequencial no final do nome, no formato nome original + número. Por exemplo, se dois pontos fossem originalmente

denominados "Parada A", o segundo seria renomeado para "Parada A 2", o terceiro para "Parada A 3", e assim por diante.

Visando facilitar a usabilidade e auxiliar o usuário a entender exatamente qual ponto de ônibus está listado, assim que selecionando, o sistema exibe automaticamente a localização do mesmo no mapa. Dessa forma, o usuário pode visualizar sua posição exata e evitar erros ao configurar alertas ou buscar informações sobre determinada linha.

Além disso, foi atribuído um identificador único (`id`) a cada ponto de ônibus. O `id` foi adicionado para facilitar a identificação individual dos pontos, um meio eficiente de referenciá-los em operações. Essa abordagem é particularmente útil em ambientes onde pontos de ônibus podem compartilhar nomes semelhantes ou idênticos, mas ainda assim precisam ser tratados como entidades distintas. Com o `id`, é possível, por exemplo, associar diretamente cada ponto de ônibus às suas coordenadas (`stop_lat` e `stop_lon`).

Após a padronização dos nomes e a atribuição dos identificadores, os dados foram organizados em um formato JSON para facilitar sua integração com aplicações *web*. Cada ponto de ônibus foi convertido em um dicionário contendo as chaves `stop_name`, `stop_lat`, `stop_lon` e `id`, e o resultado final foi salvo em um arquivo chamado `stops.json`. Com o JSON já tratado, ele é importado diretamente no *frontend* do sistema, permitindo seu consumo imediato como uma variável **JavaScript**. Isso eliminou a necessidade de carregamentos adicionais via requisições assíncronas.

5.2 DESENVOLVIMENTO DO *BACKEND*

O *backend* do sistema *Bus Tracker* foi projetado com uma arquitetura modular para facilitar a manutenção, escalabilidade e integração com APIs externas.

5.2.1 Estrutura

A implementação do *backend* foi estruturada em camadas e organizada em pastas específicas, com responsabilidades definidas a seguir:

- **data/**: Armazena os arquivos CSV relacionados aos pontos de ônibus, como dados originais dos pontos de ônibus, usados no tratamento de dados para alimentar o sistema.
- **db/**: contém a classe responsável por criar e configurar o banco de dados **MySQL**, incluindo a criação de tabelas e conexões (MySQL, 2001).
- **mocks/**: Abriga simulações usadas em testes, como respostas de APIs externas para cenários controlados.
- **services/**: Implementa a lógica de negócios do sistema, como:

- `bus_data_fetcher`: Integração com a API do **Data Rio** para obter dados sobre localização de ônibus em tempo real (PREFEITURA-RJ, 2024a).
 - `email_service`: Gerencia o envio de notificações por *email* aos usuários.
 - `travel_time_service`: Realiza cálculos de distância (em minutos) entre as coordenadas do ônibus e as coordenadas do ponto.
- `tests/`: Contém os testes automatizados para validar cada funcionalidade, como:
 - `test_process_stops`: Testa o pré-processamento dos dados de pontos de ônibus.
 - `test_email_service`: Realiza a montagem do *email*, a partir de um *template* criado em HTML, e o envia.
 - `test_travel_time_service`: Responsável pela precisão nos cálculos de tempo de viagem e distância.
 - `utils/`: Inclui scripts auxiliares, como:
 - `process_stops`: Realiza o tratamento dos dados de pontos de ônibus.
 - `tokenizer`: Auxilia na manipulação de *tokens* para autenticação.
 - `app.py`: Arquivo principal do *backend*, que centraliza a lógica do sistema por meio de *endpoints* **RESTful** (FIELDING, 2000) e integra as funcionalidades implementadas nos outros módulos.

5.2.2 Endpoints

Os endpoints foram centralizados no arquivo `app.py`, proporcionando organização e facilitando a manutenção do *backend*. Abaixo, estão listados os principais *endpoints*, agrupados por funcionalidade:

5.2.3 Desafios e Soluções Implementadas

- **Segurança do Usuário**: A segurança na autenticação e a proteção dos dados dos usuários são questões essenciais, solucionadas com a implementação de um sistema baseado em **JSON Web Tokens (JWT)**, utilizando a biblioteca **python-jose** para codificação e verificação dos *tokens* (PROJECT, 2024) (DAVIS, 2024). Para fortalecer a proteção das credenciais, as senhas são armazenadas com *hashing* usando **bcrypt**, dificultando ataques de força bruta (PYCA, 2024). Além disso, a comunicação entre *frontend* e *backend* ocorre exclusivamente via HTTPS, permitindo que os dados sejam transmitidos garantindo privacidade e segurança (CALLEGATI; CERRONI; RAMILLI, 2009).

Quadro 11 – *Endpoints* Implementados

| Categoria | Descrição |
|------------------------------|---|
| Gerenciamento de Usuários | POST /register/: Registra um novo usuário no sistema. Envia um <i>email</i> de verificação com um <i>token</i> para ativação da conta. |
| | GET /verify/: Verifica o <i>email</i> do usuário com base no <i>token</i> enviado anteriormente. |
| | POST /resend-verification/: Reenvia o <i>email</i> de verificação caso não tenha sido recebido ou tenha sido perdido. |
| | POST /token: Realiza o <i>login</i> , gerando um <i>token</i> JWT para autenticação. |
| | GET /users/me: Retorna informações do usuário autenticado. |
| Gestão de Alertas | GET /stops/: Retorna a lista de pontos de ônibus disponíveis no sistema. |
| | POST /stops/register/: Registra um novo alerta com informações específicas para monitoramento e notificações. |
| | GET /stops/registered/: Lista os alertas registrados por um usuário autenticado. |
| | DELETE /stops/registered/: Remove um alerta registrado. |
| Integração com APIs Externas | GET /infos/: Obtém informações em tempo real sobre linhas de ônibus e coordenadas dos ônibus da API do Data Rio (PREFEITURA-RJ, 2024a). |
| | GET /travel_times/: Calcula tempos de distâncias entre ônibus e um ponto de ônibus específico. |
| Redefinição de Senhas | POST /request-password-reset/: Gera um token para redefinição de senha e o envia por <i>email</i> . |
| | POST /reset-password/: Redefine a senha utilizando o <i>token</i> recebido por <i>email</i> . |

- **Falha no Cálculo de Distâncias:** A API do **OpenRouteService** (HEIGIT, 2024) apresentou dificuldades em calcular distâncias utilizando as coordenadas fornecidas pela API do **Data Rio** (PREFEITURA-RJ, 2024a). Para mitigar esse problema, foi implementado um mecanismo de *fallback* que atribui um valor padrão (-) à distância nos casos em que os dados não estão disponíveis. Essas falhas podem ser atribuídas a dois principais fatores: restrições de roteamento, decorrentes da ausência de informações detalhadas sobre vias ou de uma conectividade inadequada nos locais fornecidos, e limitações do modelo de roteamento, relacionadas a possíveis falhas nos algoritmos internos da API em condições específicas, como discrepâncias em elevações, inconsistências nos dados geográficos ou informações desatualizadas (FORUM, 2023).
- **Validação de Dados:** O uso de **pydantic** (COLVIN et al., 2024) garantiu que apenas dados válidos fossem processados, enquanto o módulo `utils/tokenizer.py` gerenciou tokens de autenticação e redefinição.
- **Ônibus passou do ponto:** A API do **OpenRouteService** (HEIGIT, 2024) não possui um método nativo para determinar se um ônibus está se aproximando ou já passou do ponto designado, pois foi projetada para calcular apenas distâncias entre coordenadas, sem levar em conta as rotas específicas dos ônibus. Isso pode gerar

alertas indevidos quando o ônibus já passou do ponto mas ainda está dentro do limite de tempo configurado pelo usuário. Para mitigar esse problema, foi desenvolvida uma heurística baseada na análise de três posições sequenciais do ônibus. Se a distância em relação ao ponto estiver diminuindo ao longo dessas posições e permanecer dentro do tempo limite configurado, considera-se que o ônibus está se aproximando. Por outro lado, se a distância estiver aumentando, mesmo dentro do tempo limite, conclui-se que o ônibus está se afastando. Essa solução garante que o usuário não receberá alertas caso o ônibus tenha passado do seu ponto.

5.3 BANCO DE DADOS

O banco de dados do *Bus Tracker* foi projetado para suportar as funcionalidades de registro de usuários, monitoramento de pontos de ônibus e gerenciamento de notificações. Ele é implementado utilizando **MySQL**, como mencionado na seção anterior (MYSQL, 2001). O esquema do banco de dados é representado na Figura 10.



Figura 10 – Esquema do Banco de Dados

5.3.1 Estrutura

A estrutura do banco de dados foi dividida em duas tabelas principais:

- **Tabela users:** Responsável por armazenar informações dos usuários do sistema, incluindo:
 - **email:** Identificador único do usuário (chave primária).
 - **hashed_password:** Senha criptografada utilizando o algoritmo **bcrypt** (PYCA, 2024).
 - **is_verified:** Indica se o *email* do usuário foi verificado.
 - **verification_token:** *Token* utilizado para a verificação do *email*.

- `reset_token`: *Token* para redefinição de senha.
- **Tabela alerts**: Gerencia os dados das paradas de ônibus cadastradas pelos usuários, contendo:
 - `email`: Identificador do usuário que cadastrou a parada.
 - `linha`: Linha de ônibus associada à parada.
 - `stop_name`: Nome da parada de ônibus.
 - `latitude` e `longitude`: Coordenadas geográficas da parada.
 - `start_time` e `end_time`: Intervalo de horário em que o usuário deseja monitorar a parada.
 - `max_distance`: Distância máxima configurada para notificações.

As tabelas foram criadas utilizando comandos SQL, para melhor integridade e a consistência dos dados.

5.3.2 Funcionalidades

A classe **BusStopDatabase** implementa diversas funcionalidades para gerenciar os dados do banco, organizadas da seguinte forma:

- **Criação do Banco de Dados e Tabelas:**
 - O método `_create_database()` verifica se o banco de dados *bustracker* já existe e o cria, caso necessário.
 - O método `_create_tables()` é responsável por criar as tabelas `users` e `alerts` com a estrutura definida, caso ainda não existam.
- **Gerenciamento de Usuários:**
 - Métodos como `register_user()` e `verify_user()` lidam com o registro e a verificação de contas de usuários.
 - Métodos para manipulação de tokens (`set_reset_token()`, `clear_reset_token()` e `get_user_by_reset_token()`) foram incluídos para suportar a redefinição de senhas.
- **Gestão de Alertas:**
 - O método `insert_alert()` registra um novo alerta associado a um usuário.
 - O método `get_alerts_by_user()` recupera todos os alertas cadastrados por um usuário específico.
 - A exclusão de alertas é tratada pelo método `delete_stop()`.

- **Consultas Gerais:**

- Métodos como `get_all_bus_stops()` e `get_all_users()` permitem listar todos os registros das tabelas `alerts` e `users`, respectivamente.

5.4 DESENVOLVIMENTO DO *FRONTEND*

O *frontend* do sistema *Bus Tracker* foi desenvolvido com uma arquitetura baseada em componentes reutilizáveis e responsivos, buscando escalabilidade e uma interface clara para os usuários.

5.4.1 Estrutura

A implementação do *frontend*, assim como no *backend*, foi organizada em pastas e arquivos específicos, com o mesmo objetivo:

- **components/**: reúne os principais elementos da aplicação, organizados para atender às diversas funcionalidades do sistema. Os componentes relacionados ao formulário, gerenciam a criação de alertas com validações integradas por meio do **React Hook Form** e da biblioteca **Zod**, permitindo a coleta de informações como linha de ônibus, ponto de ônibus, intervalo de horário e distância (em minutos) para o alerta (REACT-HOOK, 2024) (MCDONNELL, 2024). A barra lateral, projetada para facilitar a navegação, exibindo funcionalidades como a visualização de alertas cadastrados e o acesso às instruções de como utilizar o sistema. O mapa interativo, por sua vez, fornece a visualização atual da localização dos ônibus.
 - **ui/**: Para manter uma interface consistente e acessível, a pasta conta com a integração de componentes de interface pré-prontos provenientes do **ShadCN**, como botões, *sliders* e *inputs* (SHADCN, 2024). Esses elementos foram customizados para manter a identidade visual do sistema e melhorar a experiência de usuário.
- **hooks/**: Inclui a parte de adaptação da interface para dispositivos móveis, promovendo responsividade.
- **images/**: Armazena ícones e imagens utilizadas na interface, como ícones de ônibus, paradas de ônibus, além de ícones de botões para ações.

5.4.2 Funcionalidades

A aplicação conta com uma série de funcionalidades interativas e otimizadas:

- **Formulário de Alertas:** Permite o registro de alertas com validações em tempo atual. Inclui:
 - Campos validados com **Zod**, para consistência no formato e nos limites de horário (MCDONNELL, 2024).
 - Integração com um *slider* para configurar a distância máxima em minutos.
 - Suporte a seleção de paradas de ônibus através de um *dropdown* otimizado para grandes volumes de dados.
- **Mapa Interativo:** Fornece visualização em tempo real das localizações de ônibus, integrado com paradas selecionáveis.
- **Tema Personalizável:** Implementado por meio do `mode-toggle.tsx`, que permite alternância entre temas claro e escuro, melhorando a acessibilidade visual.
- **Sistema de Navegação:** A barra lateral (`app-sidebar.tsx`) possibilita fácil acesso às funcionalidades, como exibição de alertas cadastrados e instruções do sistema.
- **Integração com o *backend*:** O formulário faz chamadas para endpoints **RESTful**, como registro de alertas e busca de dados de localização de ônibus, utilizando `fetch` e atualizações de estado para refletir as informações.

5.4.3 Abordagem de Desenvolvimento

Para o gerenciamento de estados individuais, foi utilizada a biblioteca **React Hook Form**, que, em conjunto com a **Context API**, serviu como uma validação eficiente de formulários e a gestão de funcionalidades importantes, como autenticação e alternância de temas (REACT-HOOK, 2024) (META, 2024). Essas ferramentas foram complementadas por componentes acessíveis e reutilizáveis, como os provenientes do **Radix UI**, que promoveram a construção de formulários consistentes e de fácil usabilidade (WORKOS, 2024).

Foram realizados testes de responsividade em dispositivos móveis e *desktops*, para confirmar que a aplicação ofereça uma experiência fluida e adaptável a diferentes tamanhos de tela e condições de uso. Para lidar com o grande volume de dados no *dropdown* de seleção de pontos de ônibus, técnicas, como *lazy loading* (O’RINBOEV, 2023) e virtualização (onde apenas as linhas visíveis são carregadas) (VAUGHN, 2024), foram empregadas, garantindo o carregamento rápido e eficiente.

Além disso, a modularidade foi priorizada ao longo de todo o processo de desenvolvimento, com componentes bem definidos e reutilizáveis que simplificaram a continuidade e evolução da aplicação. Essa abordagem resultou em um *frontend* robusto, responsivo e alinhado às necessidades do sistema, para que tanto usuários cadastrados quanto visitantes tivessem uma interação eficiente com o *Bus Tracker*.

5.5 DEPLOY

O processo de *deploy* do *Web App* foi realizado utilizando diversas ferramentas e serviços que garantem o funcionamento da aplicação.

O domínio principal, `bustracker.com.br`, foi adquirido no **registro.br**, uma plataforma confiável para registro de domínios no Brasil. A configuração dos DNS (*Domain Name System*) foi realizada através do painel de controle do serviço. O DNS é um sistema que traduz nomes de domínio, como `bustracker.com.br`, em endereços IP utilizados pelos servidores na comunicação entre dispositivos. Ele age como uma espécie de "agenda telefônica" da Internet, permitindo que os usuários acessem *sites* usando nomes fáceis de lembrar em vez de endereços IP numéricos (REGISTRO.BR, 2024) (Cloudflare, 2024).

O *email* transacional, `info@bustracker.com.br`, utilizado para notificações automáticas da aplicação, foi configurado usando o **Zoho Email**, que oferece uma solução de fácil integração para gestão de *emails* (MAIL, 2024).

A aplicação foi hospedada em uma máquina virtual provisionada pela **Vultr**, que oferece recursos escaláveis e configuráveis para atender às necessidades do projeto. Na máquina virtual, foi utilizada a containerização por meio do **Docker** e **Docker Compose**, permitindo uma fácil gestão e implantação dos serviços do sistema. Além disso, a utilização do **Docker** permitiu a implementação de um mecanismo de recuperação automática, garantindo a restauração do serviço em caso de falhas na máquina hospedada (DOCKER, 2024b) (DOCKER, 2024a).

Para o gerenciamento de HTTPS e *proxy* reverso, foi utilizado o **Caddy**, uma ferramenta conhecida por sua simplicidade e suporte nativo a certificados SSL. A configuração incluiu redirecionamento automático para HTTPS, garantindo segurança no tráfego de dados (SERVER, 2024).

5.6 DA PROTOTIPAGEM AO CÓDIGO

A transição do design para a implementação do sistema *Bus Tracker* foi conduzida de maneira iterativa, preservando a fidelidade ao design original, apresentado no capítulo anterior. Os protótipos desenvolvidos no **Figma** serviram como base para a criação de componentes React reutilizáveis, estilizados com **TailwindCSS** e **Material-UI** (FIGMA, 2024b) (META, 2024) (TAILWIND, 2024) (MATERIAL-UI, 2024). Esse processo priorizou a organização e a consistência visual, resultando em uma interface funcional e robusta.

Os componentes foram decompostos em elementos menores e modulares, como *inputs*, formulários, tabelas e botões, com hierarquias bem definidas que facilitaram tanto o desenvolvimento quanto a manutenção. O **ShadCN** desempenhou um papel essencial nesse processo, oferecendo componentes pré-construídos estilizados com **TailwindCSS**, como menus suspensos, botões, *modals* e *sidebars*. Além disso, sua flexibilidade de personaliza-

ção permitiu adaptações às necessidades específicas do sistema, sem comprometer o seu *design*.

Para estilização e responsividade, o **TailwindCSS** foi utilizado para aplicar estilos rápidos e consistentes, enquanto o **Material-UI** complementou com componentes prontos, como *dialogs* e tabelas, promovendo acessibilidade e um *design* moderno (TAILWIND, 2024) (MATERIAL-UI, 2024).

Os componentes foram conectados ao estado global da aplicação por meio do **Redux Toolkit**, simplificando o gerenciamento de estados complexos (REDUX, 2024).

5.6.1 Principais Desafios e Soluções

O processo de tradução dos designs para o código envolveu uma série de desafios, que foram abordados com as seguintes estratégias:

- **Alinhamento entre Design e Implementação:** A fidelidade ao *design* original foi um dos principais desafios. Para isso, o recurso de inspeção de estilos do **Figma** foi amplamente utilizado, permitindo extrair propriedades detalhadas, como espaçamentos, tamanhos, cores e fontes (FIGMA, 2024b). Essa abordagem resultou na reprodução precisa dos elementos visuais e a rápida identificação de discrepâncias entre *design* e implementação.
- **Manutenção da Consistência Visual:** A ausência inicial de um *design system* aumentava o risco de inconsistências na interface. Como solução, foi desenvolvido um *design system* simplificado contendo paletas de cores, estilos tipográficos, espaçamentos padrões e componentes reutilizáveis. Esse sistema funcionou como um guia central, promovendo uniformidade e reduzindo o esforço de desenvolvimento (FIGMA, 2024a).
- **Desempenho em Mapas com Muitos Dados:** A renderização de dados de grande volume em mapas interativos gerava problemas de desempenho perceptíveis com o **Leaflet.js**, devido à quantidade de pontos de ônibus (mais de 7500), resultando no travamento da aplicação (AGAFONKIN, 2024). Após análises de desempenho, o **OpenLayers.js** foi escolhido como alternativa. Essa migração proporcionou renderizações mais rápidas e interações mais suaves, além de um suporte nativo para gerenciamento de grandes conjuntos de dados geoespaciais (OPENLAYERS, 2024).

5.7 RESULTADO DAS TELAS DO SISTEMA

Com as etapas descritas acima, o *Bus Tracker* se tornou acessível, seguro e integrado para fácil utilização, durante o período de testes, podendo ser acessado diretamente através do link: <https://bustracker.com.br/>. As telas desenvolvidas para o sistema *Bus Tracker*,

bem como os *emails* enviados aos usuários estão disponíveis no Apêndice C. As interfaces foram projetadas para dispositivos móveis e *web*, abrangendo a tela inicial, o rastreamento de ônibus em tempo real e o gerenciamento de alertas. Além disso, inclui os *emails* de alerta e de verificação de conta.

6 AVALIAÇÃO

Neste capítulo, é abordada a análise do desempenho e da usabilidade do *Bus Tracker*. Primeiramente, são descritos os testes de carga e desempenho realizados para avaliar a escalabilidade do sistema e os impactos da dependência da API do **Data Rio**. Em seguida, as estratégias adotadas para otimização do tempo de resposta, com destaque para a implementação de mecanismos de *cache*. Por fim, são apresentados os *feedbacks* dos usuários, incluindo aspectos positivos, sugestões de melhorias e possíveis aprimoramentos.

6.1 ANÁLISE DE DESEMPENHO

O sistema implementado depende de uma API externa para fornecer dados essenciais ao seu funcionamento. Essa dependência influencia diretamente o tempo de resposta e a escalabilidade do serviço. Durante os testes de desempenho, foi avaliado o tempo médio de requisição, a variação sob diferentes cargas de usuários e os impactos da latência externa. Os resultados demonstram que, à medida que a concorrência aumenta, há uma degradação perceptível na experiência do usuário, especialmente quando há restrições impostas pela API do **Data Rio** (PREFEITURA-RJ, 2024a).

Para minimizar esses impactos, a implementação de um mecanismo de *cache* se tornou essencial. Esse *cache* permite armazenar respostas frequentemente acessadas, reduzindo a necessidade de requisições repetitivas à API externa. Dessa forma, o sistema pode responder mais rapidamente às solicitações dos usuários, minimizando a sobrecarga e os tempos de espera.

6.1.1 Testes de Desempenho

O desempenho de uma aplicação *web* é um fator determinante para a retenção de usuários. Estudos apontam que a probabilidade de um usuário abandonar uma página aumenta em 32% quando o tempo de carregamento passa de 1 para 3 segundos (GOOGLE, 2017). Assim, é altamente recomendável que a resposta da interface ocorra em, no máximo, 0.4 segundos, garantindo uma experiência fluida e satisfatória para o usuário (UX, 2021).

Para avaliar a capacidade de resposta do sistema sob diferentes cargas, foi utilizada a ferramenta **Grafana k6**, empregada em testes de carga e desempenho (Grafana Labs, 2024). Essa ferramenta foi escolhida por suas características essenciais para um ambiente de produção:

- **Simplicidade na configuração:** permite definir cenários de teste de maneira declarativa e reproduzível.

- **Baixo consumo de recursos:** sua arquitetura otimizada possibilita a simulação de milhares de requisições simultâneas sem sobrecarregar a máquina de testes.
- **Relatórios detalhados:** fornece métricas precisas sobre tempo de resposta, taxa de erro e comportamento sob carga crescente.
- **Extensibilidade:** suporte a extensões e integrações com diversas ferramentas, permitindo a personalização conforme as necessidades específicas do projeto.

Para avaliar a robustez e a escalabilidade do sistema, foi conduzido um teste de carga no *endpoint* `GET /infos/`, que é o ponto mais sensível da aplicação. Esse *endpoint* é responsável por mostrar a localização dos ônibus no mapa em tempo real e também pelo disparo dos alertas para os usuários cadastrados. Como a funcionalidade depende diretamente da API do **Data Rio**, seu desempenho influencia significativamente a usabilidade do sistema. O objetivo é que o tempo de resposta deste *endpoint* seja igual ou inferior a média do **Moovit**.

O teste simulou até 1000 usuários acessando o sistema, aumentando 100 usuários a cada 2 minutos. Este experimento foi executado 5 vezes, com um tempo médio de resfriamento de 20 minutos, para evitar danificar e sobrecarregar o computador. Em cada etapa, foi monitorado o tempo de cada requisição para calcular o tempo médio e o desvio padrão, visando avaliar o desempenho do sistema sob carga crescente e identificar gargalos na comunicação com a API externa. A Figura 11 destaca os tempos de resposta atingidos, em média, quase 30 segundos em alguns casos, com grande variabilidade possivelmente devido à dificuldade do programa em alocar recursos como CPU e memória, ou à atuação do *Garbage Collector* do Python (ROSSUM; DRAKE, 2009).

Embora o número de usuários aumentasse, observou-se que o tempo de resposta não cresceu linearmente. Isso pode ser atribuído às requisições aleatórias realizadas nos testes, onde o tempo de resposta varia conforme a linha de ônibus solicitada; algumas linhas possuem mais ônibus em circulação do que outras. Portanto, picos de tempo podem ocorrer quando são solicitadas linhas com mais ônibus, enquanto solicitações de linhas com menos ônibus podem resultar em tempos de resposta menores. Os resultados destacam o impacto significativo da dependência dessa API no tempo de resposta, indicando a necessidade de otimizações para garantir maior eficiência.

Uma das principais melhorias implementadas foi a introdução de um mecanismo de *cache*, tanto manual quanto utilizando o **FastAPI-Cache** (FASTAPI-CACHE, 2024). O *cache* manual foi desenvolvido com um dicionário **Python** para armazenar os dados das linhas de ônibus, acompanhado de um timestamp que registra a última atualização. Esse mecanismo permite verificar a validade dos dados antes de realizar novas requisições, garantindo que apenas informações expiradas sejam atualizadas. Foi adotado um tempo de *cache* de 3 minutos, durante o qual todas as informações necessárias (horário

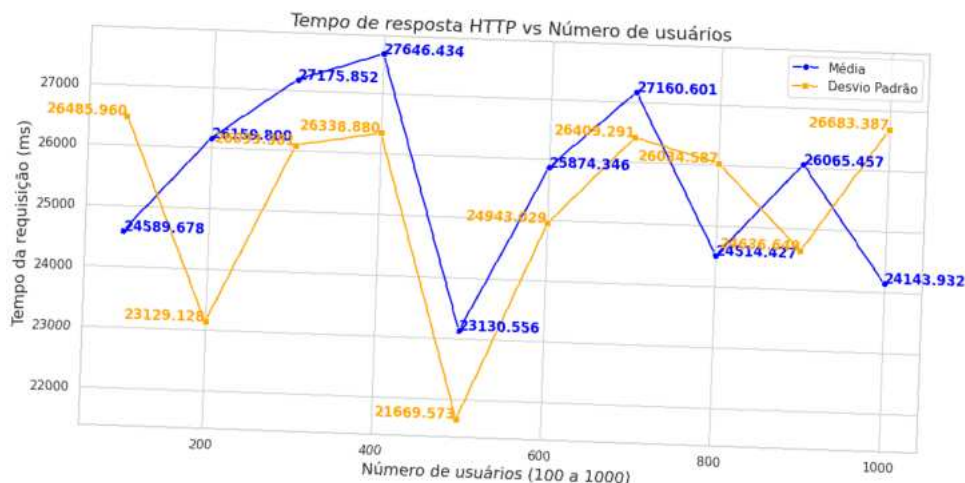


Figura 11 – Tempo de Resposta HTTP sem *cache* vs Número de Usuários

e coordenadas) permanecem armazenadas. Caso uma nova requisição seja feita dentro desse intervalo e os dados ainda sejam válidos, o sistema retorna as informações em *cache*, evitando uma nova requisição e o consequente processamento da resposta. Apesar da melhoria no tempo de resposta, essa técnica pode não fornecer dados precisos em tempo real. No entanto, sem sua aplicação e com poucos recursos, o tempo de resposta se torna inviável, como demonstrado na Figura 13.

A segunda abordagem envolveu a utilização da biblioteca **FastAPI-Cache**, que integrou caching em nível de *endpoint* por meio de decoradores, facilitando a gestão e expiração automática das respostas armazenadas (FASTAPI-CACHE, 2024). Com isso, houve uma redução expressiva no tempo médio de resposta, tornando o sistema consideravelmente mais responsivo e menos suscetível a variações de latência da API externa.

Esse tipo de otimização não apenas melhora a experiência do usuário, mas também reduz a carga sobre os servidores e melhora a escalabilidade da aplicação. Portanto, a adoção de estratégias como *caching* são essenciais para garantir um desempenho consistente e satisfatório (AWS, 2024).

Conforme ilustrado na Figura 13, a aplicação dos métodos de *caching* (manual e *endpoint*) reduziu significativamente o tempo de resposta da API, mesmo com um número relativamente alto de usuários. Em média, o tempo de resposta sem *cache* foi de aproximadamente 26.000 ms, enquanto com *cache* caiu para cerca de 370 ms, resultando em uma redução de quase 7.000% no tempo médio de resposta.

Além disso, uma abordagem promissora para aumentar a eficiência do sistema é o uso de modelos preditivos. Técnicas de aprendizado de máquina podem antecipar os dados necessários com base em padrões históricos, reduzindo a latência e otimizando as consultas à API. Esse tipo de predição já é amplamente utilizado em sistemas de mapas, como o **Google Maps**, para estimar condições de tráfego e sugerir rotas otimizadas. Embora essa estratégia possa aprimorar a precisão e o tempo de resposta da API, sua

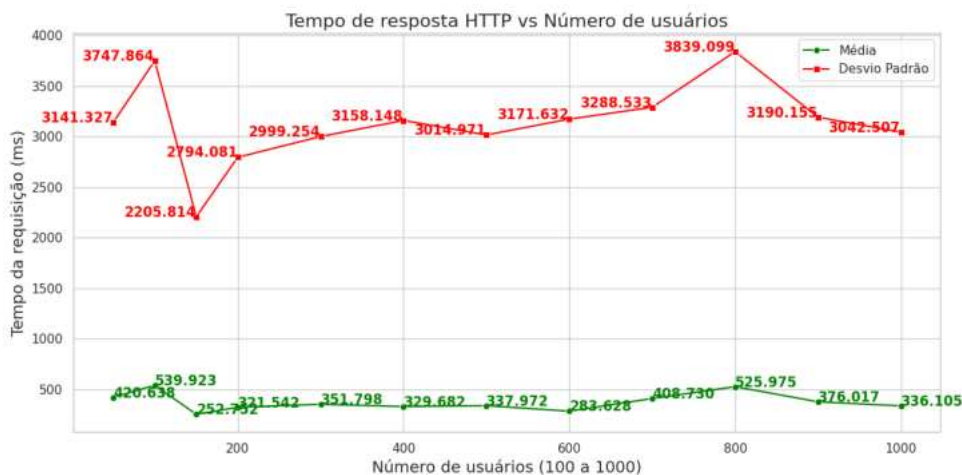


Figura 12 – Tempo de Resposta HTTP com *cache* vs Número de Usuários

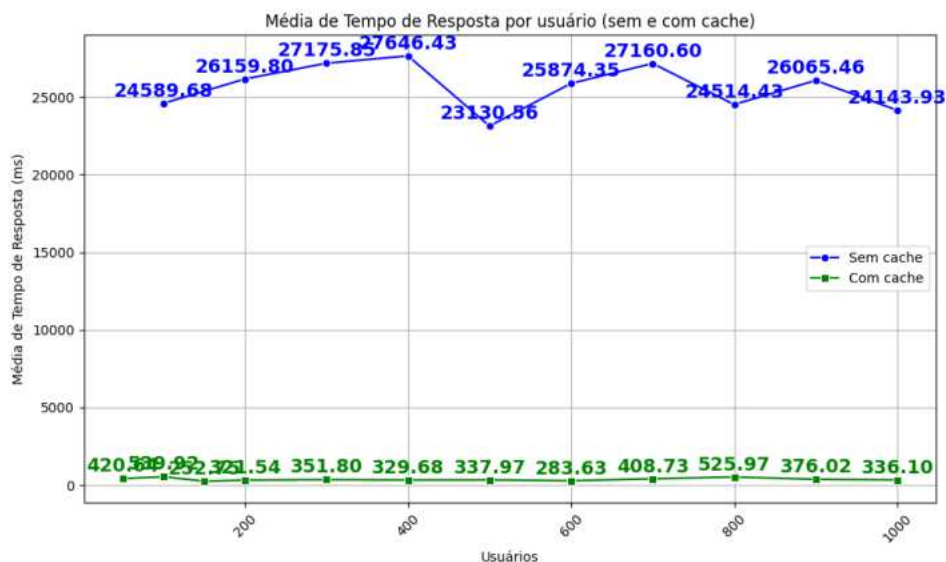


Figura 13 – Comparação da média do Tempo de Resposta HTTP com *cache* e sem *cache* por Número de Usuários

implementação exige um alto custo computacional, tornando-se inviável em máquinas que não sejam servidores equipados com GPUs e múltiplos processadores (Google Maps Team, 2020).

6.2 COMPARAÇÃO COM OUTRO APLICATIVO

Os aplicativos mencionados no Capítulo 2 não oferecem APIs públicas gratuitas, impossibilitando testes detalhados de desempenho. Para contornar essa limitação, utilizou-se o **Chrome DevTools**, um conjunto de ferramentas que permite editar páginas dinamicamente e diagnosticar problemas (GOOGLE, 2025), para medir o tempo de resposta das requisições. O **Moovit** (MOOVIT, 2024) foi escolhido para comparação devido à

funcionalidade de rastreamento de ônibus em tempo real, semelhante à do *Bus Tracker*.

No teste, avaliou-se a função que retorna o tempo estimado de chegada do ônibus mais próximo em uma determinada localidade. Após 10 repetições, obteve-se uma média de aproximadamente 850 ms para o **Moovit** (MOOVIT, 2024) e 23.000 ms para o *Bus Tracker*.

Aplicando a abordagem de *cache* descrita na seção anterior e repetindo o teste, o tempo médio do *Bus Tracker* caiu para 392 ms. No entanto, a API do **Moovit** (MOOVIT, 2024) também retorna informações adicionais, como múltiplas linhas e rotas otimizadas que combinam diferentes meios de transporte, o que pode impactar seu tempo de resposta. Assim, ao focar apenas na previsão de chegada dos ônibus, é provável que o tempo real da API do **Moovit** (MOOVIT, 2024) seja igual ou inferior à média observada no teste.

6.3 FEEDBACK DOS USUÁRIOS

Durante o período de teste de duas semanas do *Bus Tracker*, foi utilizado um formulário de avaliação para coletar as impressões dos usuários. As perguntas presentes no formulário estão disponíveis no Apêndice D. Essa escolha se deu por fatores como limite geográfico, uma vez que o pesquisador estava residindo em outra cidade, e para minimizar vieses nas respostas. O link para os testes da aplicação foi divulgado para um número reduzido de voluntários, em função dos limites de requisições das APIs utilizadas, em especial a API do Openrouteservice, que impõe um número máximo de 2.000 requisições diárias na modalidade gratuita, sendo um fator limitante. Foram obtidas 14 respostas, e o feedback demonstrou que a maioria dos usuários avaliou sua experiência com o aplicativo como "Excelente", elogiando a interface amigável, a facilidade de uso e funcionalidades úteis, como os alertas e o mapa em tempo real. Apenas um usuário classificou a experiência como "Boa", indicando uma percepção majoritariamente positiva. As respostas dos usuários foram disponibilizadas no Apêndice E.

No formulário foram destacados aspectos positivos como:

- **Alertas e notificações por email:** A praticidade de receber notificações automáticas poupando o esforço de monitoramento constante.
- **Mapa interativo:** A visualização dos ônibus e pontos em tempo atual, sendo descrita como prática e intuitiva.
- **Design e interface:** Muitos destacaram o visual atraente do aplicativo, incluindo a opção de tema claro/escuro e os *templates* de email bem elaborados.
- **Tabela de informações:** A funcionalidade que exibe a distância em minutos até o ponto foi mencionada como "genial".

6.3.1 Pontos de Melhoria e Problemas Técnicos

Alguns pontos de melhoria também foram mencionados, sendo eles:

- **Atualização em tempo real:** Alguns usuários sugeriram que o mapa poderia ser constantemente atualizado para melhorar a experiência e acompanhamento.
- **Limites de agendamento:** Foi solicitado que o intervalo máximo de agendamento fosse ampliado para além de uma hora e que fosse possível agendar alertas para dias específicos ou de forma recorrente (ex.: "dias úteis").
- **Cobertura geográfica:** Um usuário mencionou que o aplicativo se limita ao Rio de Janeiro, restringindo sua utilidade para quem reside em outras regiões.
- **Busca por linhas de ônibus:** A falta de uma funcionalidade de busca por linhas, assim como feita no *dropdown* dos pontos de ônibus.

A maioria dos usuários não encontrou erros técnicos. Houve apenas um relato isolado de informações de distância que não carregaram corretamente, mas o problema foi resolvido ao tentar novamente.

É importante destacar que todos os usuários afirmaram que recomendariam o *Bus Tracker* a outras pessoas, e a maioria indicou que o utilizaria no dia a dia. Alguns usuários mencionaram que o uso continuado dependeria de expansões funcionais ou geográficas.

6.3.2 Funcionalidades Sugeridas e Comentários Finais

Alguns usuários relataram possíveis funcionalidades que seriam úteis, caso implementadas no sistema. São elas:

- Inclusão de horários aproximados das linhas de ônibus, além do rastreamento em tempo real.
- Expansão para outros modais de transporte, como metrô, para aumentar a abrangência do aplicativo.
- Informações sobre acessibilidade, para tornar o aplicativo mais inclusivo.

Além disso, vários usuários elogiaram a iniciativa e a qualidade geral do aplicativo, enfatizando o impacto positivo que ele pode ter no cotidiano. Frases como "o trabalho ficou impecável" e "parabéns pelo esforço tanto na parte visual quanto nas ideias de funcionalidades" refletem a aprovação geral dos entrevistados.

6.4 CONFORMIDADE COM OS REQUISITOS DEFINIDOS

Ao longo deste capítulo, foi explicado como cada requisito foi atendido, detalhando as soluções adotadas para os requisitos funcionais e não funcionais. Vale ressaltar que todos os requisitos estabelecidos foram atendidos, conforme detalhado neste capítulo, com exceção da alta disponibilidade, que foi apenas parcialmente alcançada durante o período de testes.

Os requisitos funcionais foram implementados, garantindo acesso autenticado, visualização da localização dos ônibus em tempo real, configuração e gerenciamento de alertas personalizados e envio de notificações. Já os requisitos não funcionais, como desempenho sob carga, segurança dos dados por criptografia, compatibilidade com dispositivos e navegadores modernos e conformidade com a legislação de proteção de dados, também foram contemplados.

No entanto, a alta disponibilidade foi limitada pela falta de recursos para manter a infraestrutura necessária, incluindo a máquina virtual, o domínio do site e a quantidade de requisições à API do **OpenRoute** (HEIGIT, 2024). Durante os testes, o sistema operou conforme esperado, mas a ausência de investimento impede a garantia de disponibilidade constante.

Quadro 12 – Conformidade com os Requisitos Funcionais

| ID | Descrição do Requisito | Status | Comentário |
|-------------|---|--------|---|
| RQF1 | Criar conta e realizar <i>login</i> | Feito | Sistema de criptografia de senhas implementado. |
| RQF2 | Visualizar localização dos ônibus em tempo real | Feito | Integração bem-sucedida com as APIs (PREFEITURA-RJ, 2024a) (HEIGIT, 2024) disponibilizando as informações definidas no requisito. |
| RQF3 | Registrar alertas personalizados | Feito | Configuração de alertas implementada e formulário recebendo as informações corretas após o preenchimento. |
| RQF4 | Receber notificações por <i>email</i> | Feito | Envio automatizado de <i>emails</i> operante. |
| RQF5 | Gerenciar alertas cadastrados | Feito | Visualização e possibilidade de exclusão de alertas implementadas. |
| RQF6 | Pesquisar pontos de ônibus | Feito | Busca integrada a partir de botão no formato de <i>dropdown</i> possibilitando a pesquisa por nome implementada. |
| RQF7 | Consultar guia de uso do sistema | Feito | Guia detalhado e canal de contato disponíveis. |

Quadro 13 – Conformidade com os Requisitos Não Funcionais

| ID | Descrição do Requisito | Status | Comentário |
|-------|--|--------------------|---|
| RQNF1 | Processamento rápido de consultas (menos de 1 seg) | Parcialmente feito | Consultas processadas em tempo real, com tempo médio menor que 400ms apenas utilizando mecanismos de <i>cache</i> . |
| RQNF2 | Desempenho sob alta carga | Feito | Testes de <i>stress</i> confirmam desempenho estável. |
| RQNF3 | Proteção de dados sensíveis (criptografia) | Feito | Criptografia aplicada à única informação sensível manipulada pelo sistema, a senha do usuário. |
| RQNF4 | Compatibilidade com navegadores e dispositivos | Feito | Testado em Chrome, Firefox, Safari, Android e iOS. |
| RQNF5 | Recuperabilidade | Feito | Mecanismos de recuperação automática implementados a partir de definição de política utilizando o Docker Compose . |
| RQNF6 | Interface adaptável | Feito | Design responsivo para <i>desktops</i> e dispositivos móveis. |
| RQNF7 | Fácil utilização | Feito | Interface intuitiva para usuários com pouco e muito conhecimento em tecnologia. |
| RQNF8 | Alta disponibilidade | Feito | Manutenções feitas apenas fora de horário de pico garantindo disponibilidade. |

Quadro 14 – Conformidade com as Restrições

| Descrição da Restrição | Status | Comentário |
|---|--------|--|
| Conformidade com LGPD e regulamentações | Feito | Adequação às normas e informação ao usuário, no guia de uso, de que seus dados não serão utilizados para testes ou compartilhamento com terceiros. |
| Acesso exclusivo online | Feito | Sistema projetado para operação exclusiva na modalidade <i>web</i> . |
| Dados em tempo real via APIs externas | Feito | Integração com APIs (PREFEITURA-RJ, 2024a) (HEIGIT, 2024) concluída e funcional. |
| Notificações por <i>email</i> | Feito | Funcionalidade de envio de <i>emails</i> implementada para alertas, verificação de conta e redefinição de senha. |
| Foco no Rio de Janeiro | Feito | Aplicativo focado na cidade do Rio de Janeiro; expansão futura planejada. |

7 CONCLUSÃO

O desenvolvimento do *Bus Tracker* permitiu a implementação de um sistema de rastreamento de ônibus em tempo real, acessível via navegador e sem necessidade de instalação, adaptado para uso em *desktops* e dispositivos móveis. Durante a implementação, foram enfrentados desafios relacionados à confiabilidade dos dados da API do **Data Rio** (PREFEITURA-RJ, 2024a), otimização do desempenho e aprimoramento da interface.

A adoção de uma arquitetura monolítica simplificou a comunicação entre os componentes (HARRIS, 2023). Testes de usabilidade resultaram em ajustes na interface, incluindo a introdução do modo escuro e melhorias na navegação. A integração com geolocalização e notificações personalizadas possibilitou o envio de alertas configuráveis pelos usuários.

Os objetivos iniciais foram em grande parte alcançados. O sistema é gratuito e compatível com navegadores *web*, dispositivos **Android** e **iOS**, eliminando a necessidade de instalação. O rastreamento em tempo real permite o fornecimento de informações atualizadas sobre a localização dos ônibus, embora sua precisão dependa da qualidade dos dados da API utilizada, o que afeta a confiabilidade e disponibilidade das informações. A interface foi ajustada para proporcionar uma experiência simples. As notificações por *email* foram implementadas conforme planejado, permitindo alertas personalizados baseados em horários e localização. Além disso, foram adotadas medidas de segurança para proteção dos dados dos usuários, incluindo criptografia.

O *Bus Tracker* apresenta limitações em comparação a aplicativos comerciais, especialmente devido a recursos mais restritos e à ausência de suporte estruturado. No entanto, preenche lacunas ao fornecer informações objetivas integradas a alertas personalizados. Apesar dos resultados positivos nos testes iniciais, uma avaliação mais abrangente depende da ampliação da base de usuários e da coleta contínua de *feedback*.

7.1 TRABALHOS FUTUROS

O *Bus Tracker* possui grande potencial para evolução. Algumas direções para trabalhos futuros incluem:

- **Análises avançadas:** Implementação de modelos preditivos para estimar tempos de chegada com base em padrões históricos de tráfego.
- **Integração com novas APIs:** Inclusão de dados sobre congestionamento e condições meteorológicas para melhorar a precisão das previsões.
- **Expansão geográfica:** Extensão do serviço para outras cidades e estados.

- **Otimização de desempenho:** Implementação de cache dinâmico e melhoria no processamento assíncrono dos dados para reduzir tempo de resposta.
- **Personalização de alertas:** Permitir configurações mais detalhadas, como diferentes tipos de notificações e níveis de prioridade.
- **Melhoria nos dados sobre acessibilidade:** Adicionar informações sobre acessibilidade nos ônibus.

REFERÊNCIAS

- AGAFONKIN, V. **Leaflet - a JavaScript library for interactive maps**. 2024. Disponível em: <https://leafletjs.com/reference.html>.
- ALVES, R. M. et al. Development and implementation of a public transportation tracking system: Study case. **Revista Tecnologia e Sociedade**, v. 15, n. 38, p. 1–17, 2019.
- ARAÚJO, C.; SCHMIDT, S. **Em dez anos, transportes coletivos do Rio perderam um bilhão de usuários**. 2025. Disponível em: <https://oglobo.globo.com/rio/noticia/2025/01/26/em-dez-anos-transportes-coletivos-do-rio-perderam-um-bilhao-de-usuarios.ghtml>.
- AWS. **Caching on AWS**. 2024. Accessed: 2024-12-30. Disponível em: <https://aws.amazon.com/caching/>.
- BARROS-JUSTO, J. L.; BENITTI, F. B. V.; TIWARI, S. The impact of use cases in real-world software development projects: A systematic mapping study. **arXiv preprint arXiv:1906.06754**, 2019.
- BIERMAN, G.; ABADI, M.; TORGERSEN, M. Understanding typescript. In: SPRINGER. **European Conference on Object-Oriented Programming**. [S.l.], 2014. p. 257–281.
- BUSINESS, W. **Preços da Plataforma do WhatsApp Business**. 2024. Acessado em: 26 de agosto de 2024. Disponível em: https://business.whatsapp.com/products/platform-pricing?lang=pt_BR.
- CALLEGATI, F.; CERRONI, W.; RAMILLI, M. Man-in-the-middle attack to the https protocol. **IEEE Security & Privacy**, IEEE, v. 7, n. 1, p. 78–81, 2009.
- CEDER, A. Urban mobility and public transport: future perspectives and review. **International Journal of Urban Sciences**, v. 25, n. 4, p. 455–479, 2021.
- CHRISTIE, T. **Uvicorn Documentation**. 2024. Disponível em: <https://www.uvicorn.org/>.
- CITTAMOBIL. **Informações em Tempo Real sobre Transporte Público**. 2024. <https://www.cittamobi.com.br>.
- CLICKSEND. **API de Gateway de SMS**. 2024. Acessado em: 26 de agosto de 2024. Disponível em: <https://www.clicksend.com/br/sms/sms-gateway-api/>.
- Cloudflare. **What is DNS? | Cloudflare**. 2024. Disponível em: <https://www.cloudflare.com/learning/dns/what-is-dns/>.
- CNM. **Panorama Geral do Transporte Público Coletivo nos Municípios Brasileiros**. [S.l.]: Confederação Nacional de Municípios, 2024. https://cnm.org.br/storage/biblioteca/2024/Estudos_tecnicos/202403_ET_MOBI_Panorama_geral_de_Transporte_Publico_Coletivo_Municipios2.pdf.

CNT. **Pesquisa CNT de Mobilidade da População Urbana**. Brasília: Confederação Nacional do Transporte, 2024. ISBN 978-85-68865-22-4.

COLVIN, S. et al. **Pydantic**. 2024. Disponível em: <https://github.com/pydantic/pydantic>.

DAVIS, M. **python-jose Documentation**. 2024. Disponível em: <https://python-jose.readthedocs.io/>.

DOCKER. **Docker Compose Documentation**. 2024. Disponível em: <https://docs.docker.com/compose/>.

DOCKER. **Docker Documentation**. 2024. Disponível em: <https://docs.docker.com/manuals/>.

ELIJAH, O. et al. Transforming urban mobility with Internet of Things: public bus fleet tracking using proximity-based Bluetooth beacons. **Frontiers in Internet of Things**, v. 2, p. 1255995, 2023.

FASTAPI-CACHE. **FastAPI-Cache: Caching for FastAPI**. 2024. Accessed: 2024-12-27. Disponível em: <https://pypi.org/project/fastapi-cache/>.

FIELDING, R. Representational state transfer. http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, University of California, 2000.

FIGMA. **Design Systems 101: What is a design system?** 2024. <https://www.figma.com/blog/design-systems-101-what-is-a-design-system/>.

FIGMA. **Figma Learn**. 2024. Disponível em: <https://help.figma.com/hc/en-us/categories/360002042553>.

FORUM, O. **Distance Matrix Returning None Values**. 2023. <https://ask.openrouteservice.org/t/distance-matrix-returning-none-values/5194>.

GLOBO, O. **Em dez anos, transportes coletivos do Rio perderam um bilhão de usuários**. 2025. <https://oglobo.globo.com/rio/noticia/2025/01/26/em-dez-anos-transportes-coletivos-do-rio-perderam-um-bilhao-de-usuarios.ghtml>. Acessado em: 2025-02-26.

GOODWIN, M. **O que é uma API (Interface de Programação de Aplicativos)?** 2024. Disponível em: <https://www.ibm.com/br-pt/topics/api>.

GOOGLE. **Google Maps Platform Documentation**. 2024. <https://developers.google.com/maps/documentation/geolocation/usage-and-billing?hl=pt-br>.

GOOGLE. **Planejamento de Rotas e Informações de Transporte Público**. 2024. <https://maps.google.com>.

GOOGLE. **Chrome DevTools**. [S.l.], 2025. Accessed: 2025-02-18. Disponível em: <https://developer.chrome.com/docs/devtools/>.

Google Maps Team. **Google Maps 101: How AI helps predict traffic and determine routes**. 2020. Disponível em: <https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/>.

GOOGLE, T. with. Page load time statistics: The need for speed. **Think with Google**, 2017. Disponível em: <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/page-load-time-statistics/>.

Grafana Labs. **Grafana k6 Documentation**. [S.l.], 2024. Accessed: 2024-12-19. Disponível em: <https://grafana.com/docs/k6/latest/>.

HARRIS, C. Microservices vs. monolithic architecture. **Atlassian**, 2023. Disponível em: <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>.

HEIGIT. **OpenRouteService API**. 2024. Disponível em: <https://openrouteservice.org/services/>.

KREKEL, H. et al. **pytest 8.3**. 2004. Disponível em: <https://github.com/pytest-dev/pytest>.

LIU, Z. et al. A relational model for formal object-oriented requirement analysis in uml. In: **Formal Methods and Software Engineering**. [S.l.]: Springer, 2003. p. 641–664.

MAIL, Z. **Zoho Mail Help Documentation**. 2024. Disponível em: <https://www.zoho.com/mail/help/>.

MATERIAL-UI. **Material-UI Documentation**. 2024. Disponível em: <https://mui.com/material-ui/getting-started/overview/>.

MCDONNELL, C. **Zod Documentation**. 2024. Disponível em: <https://zod.dev/>.

META. **React Documentation**. 2024. Disponível em: <https://reactjs.org/docs/getting-started.html>.

MOBILIDADE, E. **Desafios do Transporte Público no Brasil**. 2023. <https://mobilidade.estadao.com.br/mobilidade-para-que/desafios-do-transporte-publico-no-brasil/>.

MONSYNE, T. D. D. et al. **smtplib — SMTP protocol client**. 2024. Disponível em: <https://docs.python.org/3/library/smtplib.html>.

MOOVIT. **Planejamento de Rotas e Rastreamento de Transporte Público**. 2024. <https://moovitapp.com>.

MYSQL, A. **MySQL reference manual**. 2001.

OLIVEIRA, L. F. S. D. **Informações em Tempo Real sobre Transporte Público em São Paulo**. 2024. <https://cadeonibus.com.br>.

OPENLAYERS. **OpenLayers Documentation**. 2024. Disponível em: <https://openlayers.org/en/latest/apidoc/>.

O’SULLIVAN, D.; MORRISON, A.; SHEARER, J. Using desktop gis for the investigation of accessibility by public transport: an isochrone approach. **International Journal of Geographical Information Science**, Taylor & Francis, v. 14, n. 1, p. 85–104, 2000.

- O'RINBOEV, A. Analyzing the efficiency and performance optimization techniques of react.js in modern web development. : , v. 2, n. 24, p. 54–57, 2023.
- PAIVA, S. et al. Enabling technologies for urban smart mobility: Recent trends, opportunities and challenges. **Sensors**, v. 21, n. 6, p. 2143, 2021.
- PANDAS. **pandas-dev/pandas: Pandas**. 2024. Disponível em: <https://github.com/pandas-dev/pandas>.
- PREFEITURA-RJ. **Data.Rio API**. 2024. Disponível em: <https://www.data.rio/documents/PCRJ::transporte-rodovi%C3%A1rio-api-de-gps-dos-%C3%B4nibus-sppo-beta/about?path=>.
- PREFEITURA-RJ. **Pontos de Ônibus - Data.Rio**. 2024. Disponível em: https://www.data.rio/datasets/fd07613c9a1c45299389c0f7cff8e2a0_0/explore.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software-9**. [S.l.]: McGraw Hill Brasil, 2021.
- PROJECT, P. **PyJWT Documentation**. 2024. Disponível em: <https://pyjwt.readthedocs.io/>.
- PYCA. **bcrypt 4.2.1**. 2024. Disponível em: <https://pypi.org/project/bcrypt/>.
- QUICKO. **Planejamento de Rotas Multimodais e Recarga de Cartões de Transporte**. 2024. <https://www.quicko.com.br>.
- RAMÍREZ, S. **FastAPI Documentation**. 2024. Disponível em: <https://fastapi.tiangolo.com/>.
- REACT-HOOK. **React Hook Form Documentation**. 2024. Disponível em: <https://react-hook-form.com/get-started>.
- REDUX. **Redux Toolkit Documentation**. 2024. Disponível em: <https://redux-toolkit.js.org/usage/usage-guide>.
- REGISTRO.BR. **Ajuda Registro.br**. 2024. Disponível em: <https://registro.br/quem-somos>.
- RÍOS, J. R. M. et al. Evaluación de los frameworks en el desarrollo de aplicaciones web con python. **Archivo de la revista Latinoamericana de Ingeniería de Software**, v. 4, n. 4, p. 201–207, 2016.
- ROMAN. A taxonomy of current issues in requirements engineering. **Computer**, IEEE, v. 18, n. 4, p. 14–23, 1985.
- ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.
- SERVER, C. **Caddy Documentation**. 2024. Disponível em: <https://caddyserver.com/docs/>.
- SHADCN. **shadcn/ui Documentation**. 2024. Disponível em: <https://ui.shadcn.com/docs>.

- TAILWIND. **Tailwind CSS Documentation**. 2024. Disponível em: <https://tailwindcss.com/docs>.
- TAUBERER, J. **email-validator 2.2.0**. 2024. Disponível em: <https://pypi.org/project/email-validator/>.
- TWILIO. **API de SMS**. 2024. Acessado em: 26 de agosto de 2024. Disponível em: <https://www.twilio.com/pt-br/messaging/channels/sms>.
- UX, A. Leis de ux: Os princípios básicos de ux design. **Medium**, 2021. Disponível em: <https://medium.com/aela/leis-de-ux-os-princ%C3%ADpios-b%C3%A1sicos-de-ux-design-9b9bf3fdb43c>.
- VAUGHN, B. **React Virtualized: React components for efficiently rendering large lists and tabular data**. 2024. Disponível em: <https://github.com/bvaughn/react-virtualized>.
- WATKINS, K. E. et al. Where Is My Bus? impact of mobile real-time information on the perceived and actual wait time of transit riders. **Transportation Research Part A: Policy and Practice**, v. 45, n. 8, p. 839–848, 2011.
- WONG, S. Y.; KOH, K. C.; CHEONG, W. H. Towards smart mobility in urban spaces: Bus tracking and information application. In: **Proceedings of the 2nd International Conference on Applied Science and Technology (ICAST 2017)**. [S.l.: s.n.], 2017. (AIP Conference Proceedings, v. 1891), p. 020145.
- WORKOS, I. **Radix UI Documentation**. 2024. Disponível em: <https://www.radix-ui.com/docs/primitives/overview/introduction>.
- ÔNIBUS, V. de. **Rastreamento e Planejamento de Transporte Público no Rio de Janeiro**. 2024. <https://www.vadeonibus.com.br>.

APÊNDICE A – FORMULÁRIO DE PESQUISA DE DEMANDA

Pesquisa: Experiências e Melhorias no Uso de Ônibus

Olá! Seja bem-vindo(a) ao formulário de pesquisa sobre o uso de ônibus no transporte público.

Esta pesquisa tem como objetivo compreender os hábitos, dificuldades e expectativas dos usuários de ônibus. Os resultados serão utilizados para o desenvolvimento do Trabalho de Conclusão de Curso no curso de Bacharelado em Ciência da Computação.

Ressaltamos que nenhuma informação que permita sua identificação será divulgada, garantindo total confidencialidade dos dados coletados.

O preenchimento deste formulário levará cerca de 5 minutos.

Sua contribuição é fundamental para o sucesso deste trabalho. Agradecemos desde já pela sua participação! :)

* Indica uma pergunta obrigatória

Você autoriza suas respostas a serem coletadas, armazenadas e analisadas? *

Sim

Não

Qual a sua faixa etária? *

- 17 anos ou menos
- 18 a 24 anos
- 25 a 34 anos
- 35 a 44 anos
- 45 a 54 anos
- 55 anos ou mais

Você utiliza ônibus como meio de transporte público? *

- Sim
- Não

Com que frequência você utiliza os ônibus? *

- Diariamente
- Algumas vezes por semana
- Semanalmente
- Ocasionalmente

Como você geralmente descobre quando um ônibus vai chegar? *

- Usando um aplicativo ou site
- Perguntando a outras pessoas
- Não verifico, apenas espero
- Outro:

Quais desafios você enfrenta ao utilizar os ônibus? *

- Nenhum
- Ônibus chegando adiantados ou atrasados
- Incerteza sobre as rotas de ônibus
- Dificuldade em saber quando o próximo ônibus chegará
- Falta de notificações sobre atrasos ou mudanças
- Outro:

Que informações sobre os ônibus seriam mais úteis para você? *

- Localização em tempo real do ônibus
- Tempo estimado de chegada
- Alertas sobre atrasos ou mudanças de rota
- Notificações quando o ônibus estiver próximo ao meu ponto
- Outro:

Você já utilizou algum aplicativo ou ferramenta para acompanhar os ônibus? *

Sim

Não

O que você gosta no aplicativo/ferramenta que utiliza?


.....

Quais recursos você sente que faltam nos aplicativos/ferramentas que já utilizou?

.....

Há mais alguma coisa que você gostaria de compartilhar sobre sua experiência com ônibus ou transporte público? Alguma outra funcionalidade que sente falta?

.....

Enviar  Página 1 de 5 [Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Instituto de Computação - UFRJ.
Does this form look suspicious? [Relatório](#)

Google Formulários

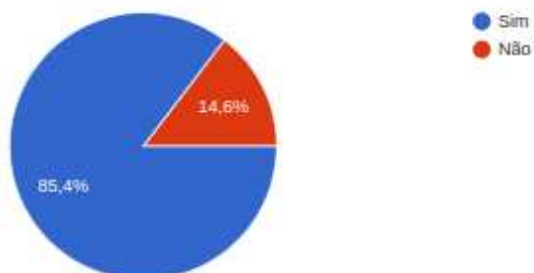
APÊNDICE B – RESULTADOS DO FORMULÁRIO DE PESQUISA DE DEMANDA



Você utiliza ônibus como meio de transporte público?

 Copiar gráfico

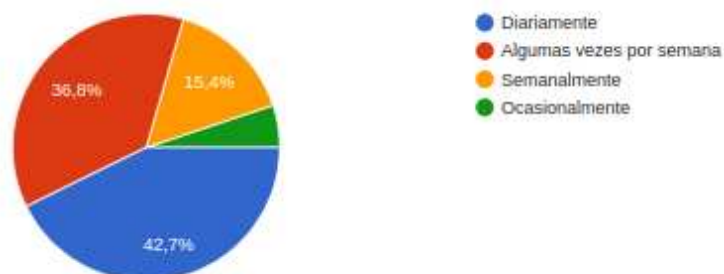
137 respostas



Com que frequência você utiliza os ônibus?

 Copiar gráfico

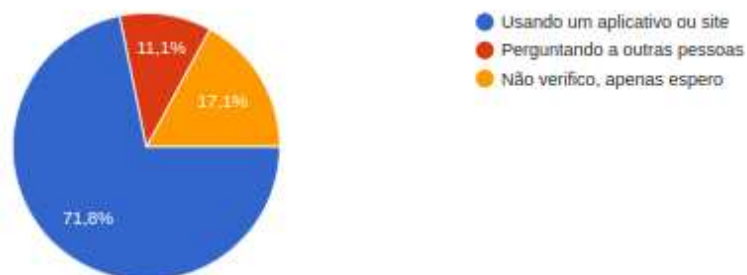
117 respostas



Como você geralmente descobre quando um ônibus vai chegar?

 Copiar gráfico

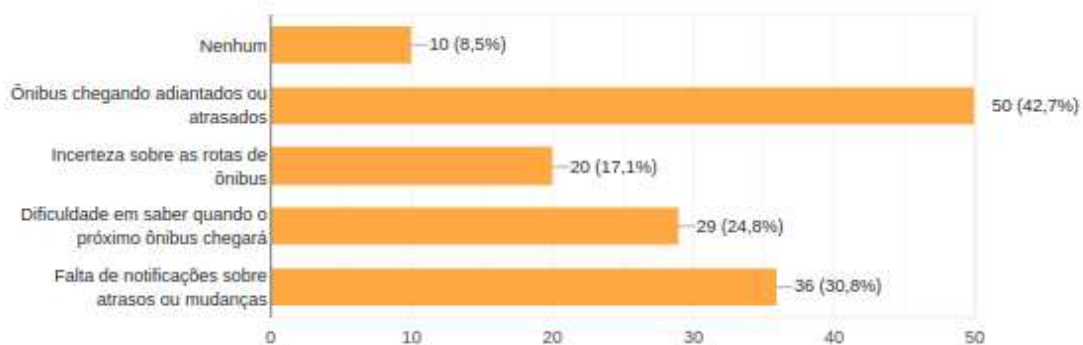
117 respostas



Quais desafios você enfrenta ao utilizar os ônibus?

[Copiar gráfico](#)

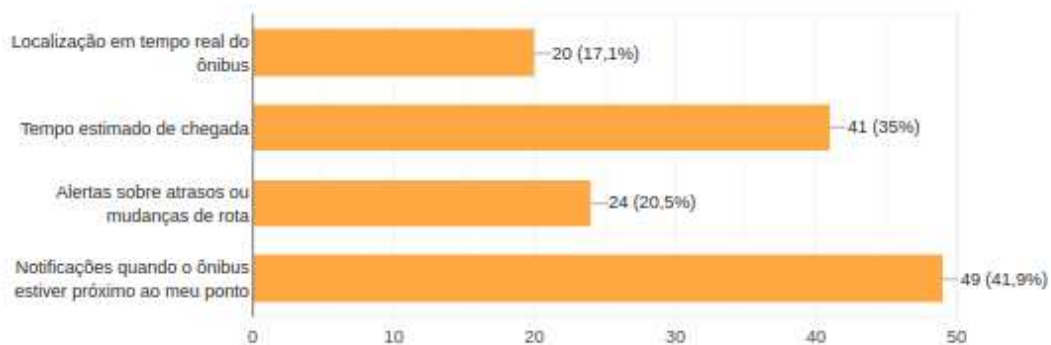
117 respostas



Que informações sobre os ônibus seriam mais úteis para você?

[Copiar gráfico](#)

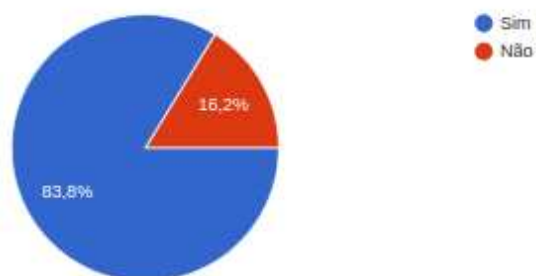
117 respostas



Você já utilizou algum aplicativo ou ferramenta para acompanhar os ônibus?

[Copiar gráfico](#)

117 respostas



| O que você gosta no aplicativo/ferramenta que utiliza? - Parte 1 |
|--|
| "app sem complicações" |
| "facilidade de uso e informações relevantes como localização" |
| "Estimativa de chegada dos ônibus, mesmo que com certo atraso de vez em quando" |
| "Facilidade de uso e design intuitivo" |
| "grande número de informações sobre as linhas que utilizo e facilidade na utilização, bem intuitivo" |
| "Localização dos ônibus em tempo real." |
| "facilidade de uso pq nao entendo muito de tecnologia, mas ainda me atrapalho um pouco. gosto da possibilidade de ver a distância do ponto de ônibus que pego" |
| "Previsão de chegada dos ônibus." |
| "Facilidade no dia a dia com dados que preciso" |
| "rastreamento de ônibus ao vivo." |
| "ampla cobertura de linhas de ônibus." |
| "Horários previstos dos ônibus." |
| "Monitoramento em tempo real dos ônibus" |
| "Rastreamento em tempo real dos ônibus." |
| "Informações úteis para o cotidiano" |
| "Múltiplas linhas disponíveis para rastreamento." |
| "interface amigável e notificações em tempo real." |
| "possibilidade de ver previsão de horários." |
| "Atualizacoes constantes dos onibus é o essencial para mim." |
| "visualização dos ônibus se aproximando do ponto" |
| "informações claras e objetivas." |
| "Informações práticas dos ônibus." |
| "layout amigável e ferramentas que preciso como a distancia." |
| "todas as linhas que preciso acompanhar estão disponíveis" |
| "Possibilidade de ver os ônibus que pego diariamente perto do ponto da minha casa para não me atrasar." |
| "Horários atualizados" |
| "horários aproximados dos ônibus" |
| "dados acessíveis para rotina diária" |
| "acho que a parte mais importante é a distancia dos onibus e visualizar no mapa pra facilitar." |
| "acompanhamento dos ônibus em tempo real (infelizmente, as vezes tem atrasos)." |
| "os horários, as linhas e a facilidade de ver onde estão os ônibus para ajudar a programar o tempo que preciso sair de casa ou do trabalho." |
| "Não curto muito o que utilizo, mas me mostra com certa facilidade os ônibus e a distância, mesmo que muitas vezes não seja preciso" |

Quadro 15 – Respostas sobre o que os usuários gostam no aplicativo/ferramenta que utilizam - Parte 1

| O que você gosta no aplicativo/ferramenta que utiliza? - Parte 2 |
|--|
| "uma estimativa de horários" |
| "monitoramento em tempo real dos ônibus, principalmente quando estou saindo do trabalho" |
| "Diversidade de linhas de ônibus disponíveis para conseguir acompanhar." |
| "tempo estimado de chegada dos ônibus" |
| "Informações úteis sobre ônibus e outros meios de transporte para o dia a dia." |
| "Navegação eficiente e funcionalidades interessantes" |
| "sei que tem outros apps que tbm tem a mesma ideia mas gosto do meu pq acho descomplicado" |
| "praticidade com informações em tempo real" |

Quadro 16 – Respostas sobre o que os usuários gostam no aplicativo/ferramenta que utilizam - Parte 2

| Quais recursos você sente que faltam nos aplicativos/ferramentas que já utilizou? - Parte 1 |
|--|
| "nada a adicionar, obrigado." |
| "não tenho muito o que sugerir, tô satisfeito" |
| "um jeito de ver as condicoes do transito, verificar o ônibus no mapa seria legal" |
| "acho que seria útil ter previsão baseada em distância e avisos sobre." |
| "sinto falta de um não precisar baixar aplicativos, prefiro sites para não precisar gastar espaço visto que já gasto internet hahah" |
| "poder visualizar o status da linha seria bacana." |
| "nao quero ter que baixar outro app" |
| "avisos de mudanças no horário são importantes" |
| "integracao com email ou whatsapp seria otimo" |
| "as vezes sinto falta de saber o tempo exato que o onibus chega." |
| "queria receber um lembrete quando tem um ônibus perto do ponto." |
| "eu queria poder customizar as notificações do meu jeito" |
| "FALTA TER ALERTA QUANDO O ONIBUS TA CHEGANDO NA PARADA." |
| "Alertas configuráveis seriam interessantes." |
| "eu queria poder configurar algum tipo de aviso de proximidade" |
| "SERIA INTERESSANTE VER INFORMAÇÕES SOBRE OS PONTOS DE ÔNIBUS, TALVEZ NOME OU NOME DA RUA EM SI PARA FACILITAR" |

Quadro 17 – Respostas sobre os recursos que faltam nos aplicativos/ferramentas utilizados - Parte 1

| Quais recursos você sente que faltam nos aplicativos/ferramentas que já utilizou? - Parte 2 |
|--|
| "curto o design atual, mas podia ser mais fácil de mexer, sinto certa dificuldade e não há lugar ou meio de tirar dúvidas." |
| "Acho que o problema mesmo é a incerteza dos horários, não sei se é apenas problema do transporte público ou do próprio aplicativo que uso." |
| "PODIA TER UMA NOTIFICACAO POR MENSAGEM EMAIL TAMBEM." |
| "mais clareza nos horários ajudaria mt" |
| "as vezes o mapa nao ajuda mt, podia ser mais detalhado com informações de distância e velocidade" |
| "seria bom se mostrasse informações sobre localização em tempo real" |
| "PODIA AVISAR SOBRE ÔNIBUS DAS LINHAS QUE SE APROXIMAM DE ALGUM LUGAR ESPECÍFICO (TALVEZ SE APROXIMANDO DE UM PONTO ESCOLHIDO)" |
| "gostaria de receber notificações quando estiver quase na hora do ônibus passar em um ponto específico (perto de onde saio do trabalho)" |
| "tudo certo, nao sinto falta de nada" |
| "SINTO A NECESSIDADE DE USAR MAIS DE UMA PLATAFORMA PRA TER AS INFORMAÇÕES QUE PRECISO." |
| "às vezes eu queria saber quanto tempo tenho até o ônibus chegar" |
| "faz falta um lugar pra visualizar os ônibus das minhas linhas" |
| "PODIA AVISAR SE O ONIBUS TA VINDO." |
| "seria otimo se mostrasse opcoes alternativas como agendar algum aviso para facilitar" |
| "poder adicionar notificações personalizadas às linhas seria incrível." |
| "CONSEGUIR VISUALIZAR TODOS OS PONTOS PRÓXIMOS DO MEU TRAJETO SERIAM ÚTEIS." |
| "uma função pra lembrar de sair de casa quando o ônibus estiver por perto seria útil" |
| "seria legal se mostrasse mapas e possibilidade de agendamento de alertas, sem necessidade de pagamento para essa funcionalidade" |
| "poder ver os ônibus que estão vindo numa lista seria bom" |
| "faz falta saber quantos minutos até a chegada exata" |
| "às vezes não confio no tempo de espera que aparece, devia ser mais certo" |
| "poderia ter a possibilidade de agendar horario pra receber notificação, e nao so favoritar a linha que uso e precisar verificar sempre. acredito que até exista isso mas pelo que vi é pago e ngm fala se é bom ou nao" |

Quadro 18 – Respostas sobre os recursos que faltam nos aplicativos/ferramentas utilizados - Parte 2

| Há mais alguma coisa que você gostaria de compartilhar sobre sua experiência com ônibus ou transporte público? Alguma outra funcionalidade que sente falta? - Parte 1 |
|--|
| "um único app que oferece planejamento completo (horário, notificações, linha, ponto, um pouco de tudo)" |
| "sistema centralizado para planejar rotas e horários" |
| "Sistema centralizado com notificações ajustáveis e localização" |
| "Notificações automáticas para distâncias específicas" |
| "todas as informações integradas em uma interface intuitiva" |
| "Dados em tempo real para planejamento do dia" |
| "Visualização clara de todas as rotas em um app" |
| "Todas as informações necessárias em um único app" |
| "opção de configurar notificações para horários específicos" |
| "notificações automáticas para localização" |
| "um único app com notificações completas" |
| "notificações ajustáveis pra facilitar o planejamento" |
| "um app único que reúne tudo sobre transporte sem distrações e com algum tipo de notificação para não ficar vendo o tempo todo a mudança da localização do ônibus" |
| "alertas personalizados para proximidade e alguma ferramenta que não precise baixar para poupar espaço no celular kkkkk" |
| "dados sobre localização e previsão de atrasos" |
| "localização, previsão e alertas" |
| "informações em tempo real sobre localização dos ônibus" |
| "Notificações personalizadas para diferentes linhas de ônibus" |
| "Previsão confiável dos horários dos ônibus" |
| "Notificações que evitam necessidade de checar o mapa" |
| "uma ferramenta que avisa sobre a proximidade dos ônibus sem precisar olhar o mapa direto" |
| "localização e planejamento em um app só com as infos dos ônibus que utilizo e se estão chegando no ponto que pego" |
| "Um app completo com alertas e previsões confiáveis" |
| "um único app com todas as informações que preciso" |
| "dados centralizados sobre atrasos, localização e horários" |
| "Dados confiáveis de localização e previsão de horários" |

Quadro 19 – Respostas sobre experiências e funcionalidades desejadas em aplicativos de transporte público - Parte 1

| Há mais alguma coisa que você gostaria de compartilhar sobre sua experiência com ônibus ou transporte público? Alguma outra funcionalidade que sente falta? - Parte 2 |
|--|
| "planejamento eficiente com informações completas" |
| "Um único app com localização e planejamento integrado" |
| "um único app que avisa sobre a localizacao dos onibus que eu uso" |
| "um app centralizado para não precisar usar vários" |
| "opção de visualizar dados em tempo real" |
| "Opção de ver a localização e distâncias em um só lugar, principalmente separando pelos ônibus que utlizo" |
| "informações atualizadas sobre atrasos" |
| "opção de configurar alertas para horários que utilizo os ônibus" |

Quadro 20 – Respostas sobre experiências e funcionalidades desejadas em aplicativos de transporte público - Parte 2

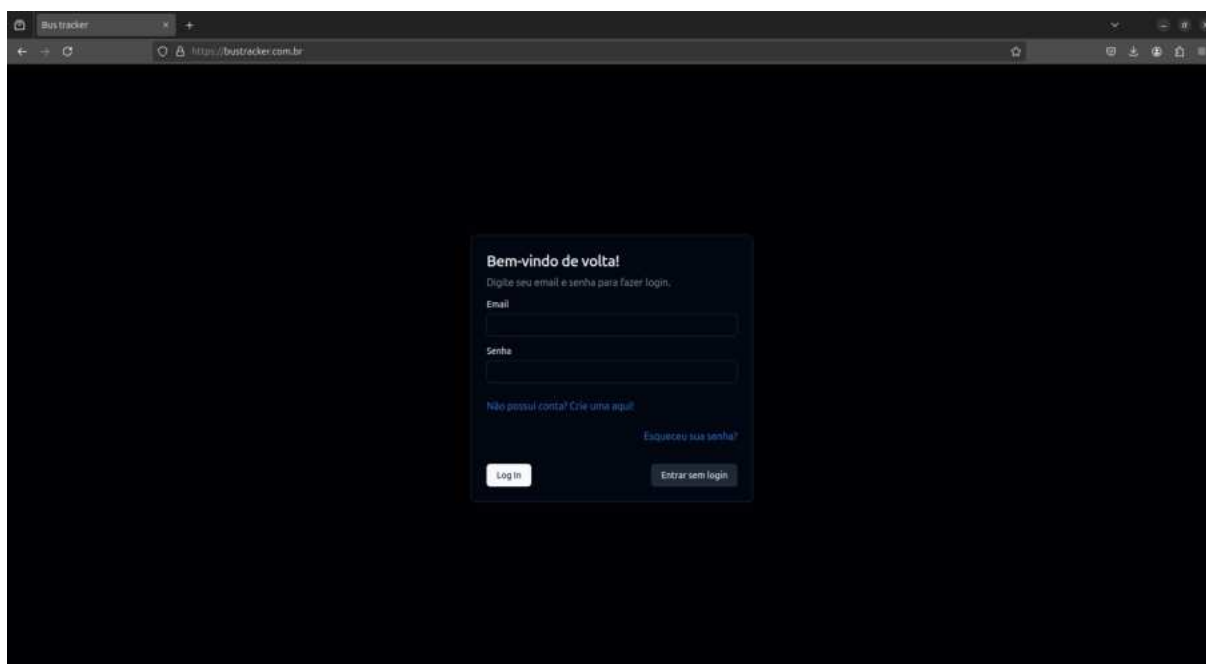
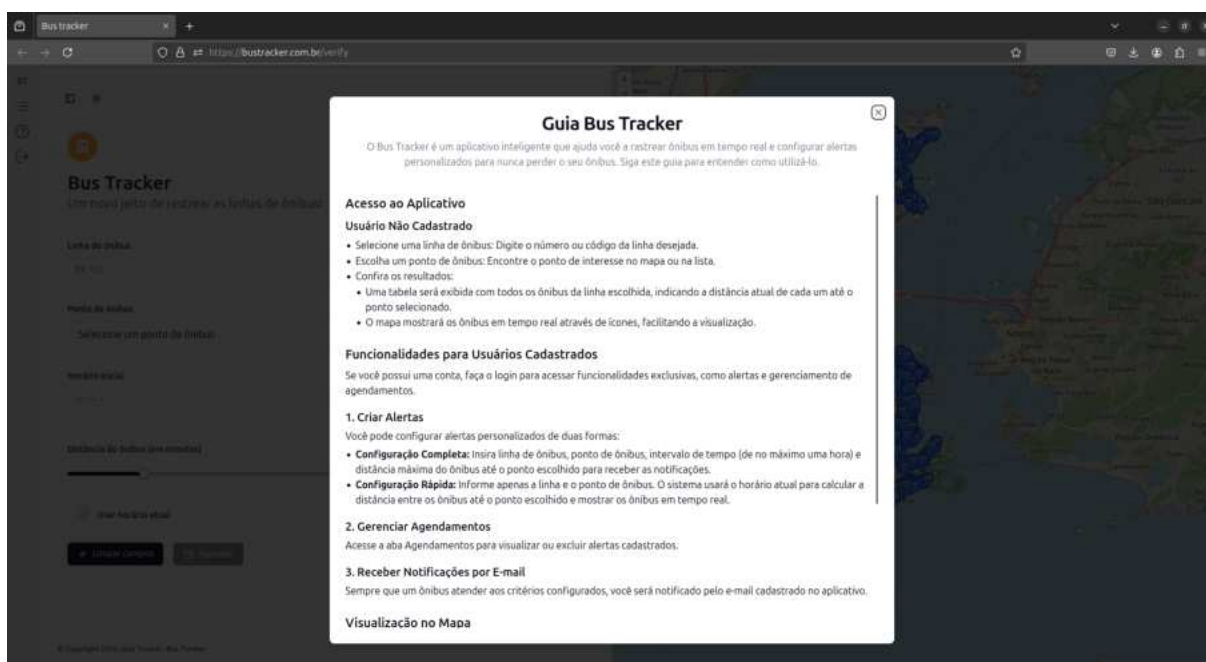
APÊNDICE C – RESULTADOS DAS TELAS DO *BUS TRACKER*

Figura 14 – Login

Figura 15 – Guia de Utilização do *Bus Tracker*

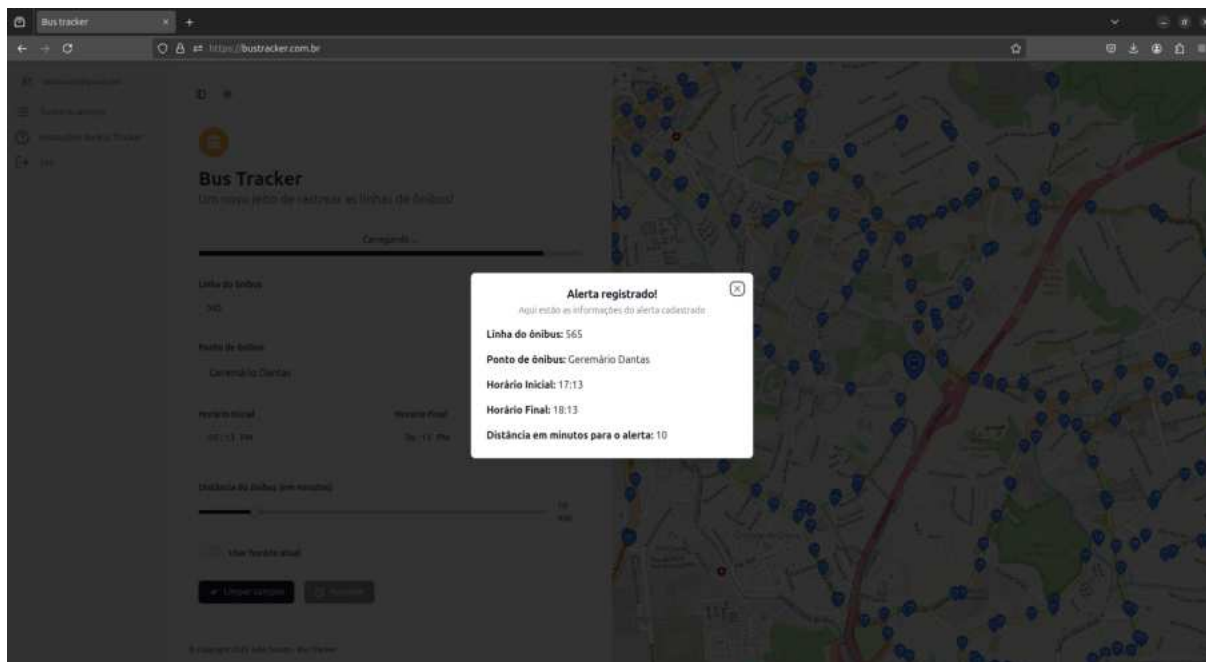


Figura 16 – Alerta Registrado

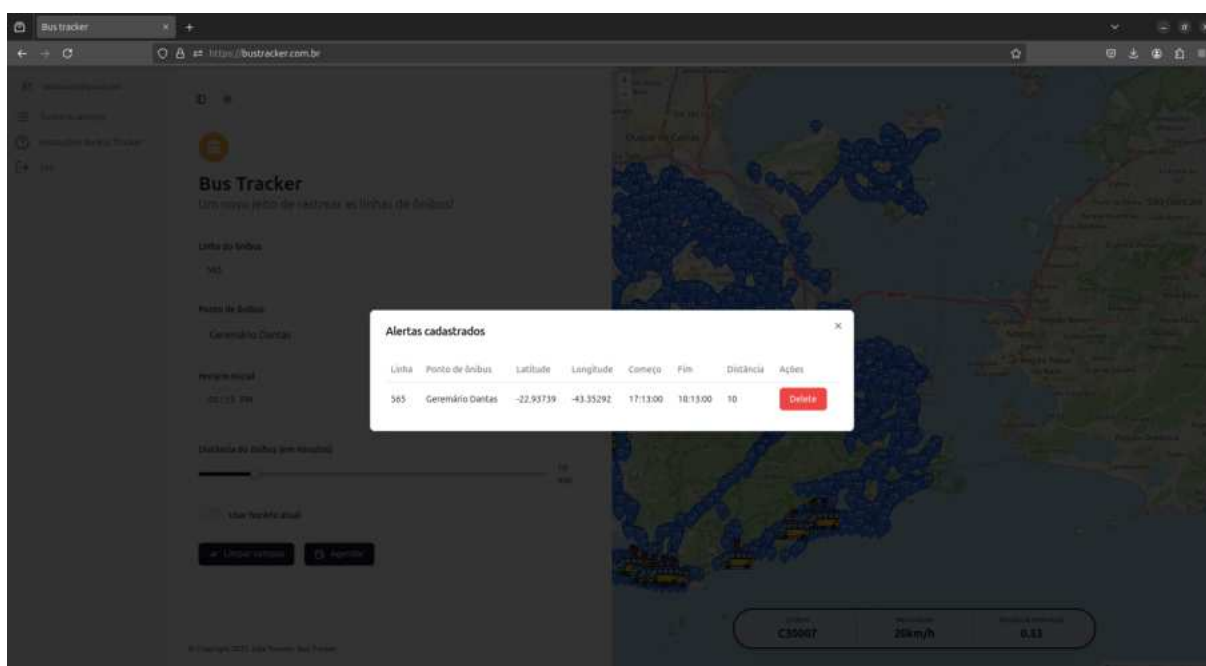


Figura 17 – Alertas Cadastrados

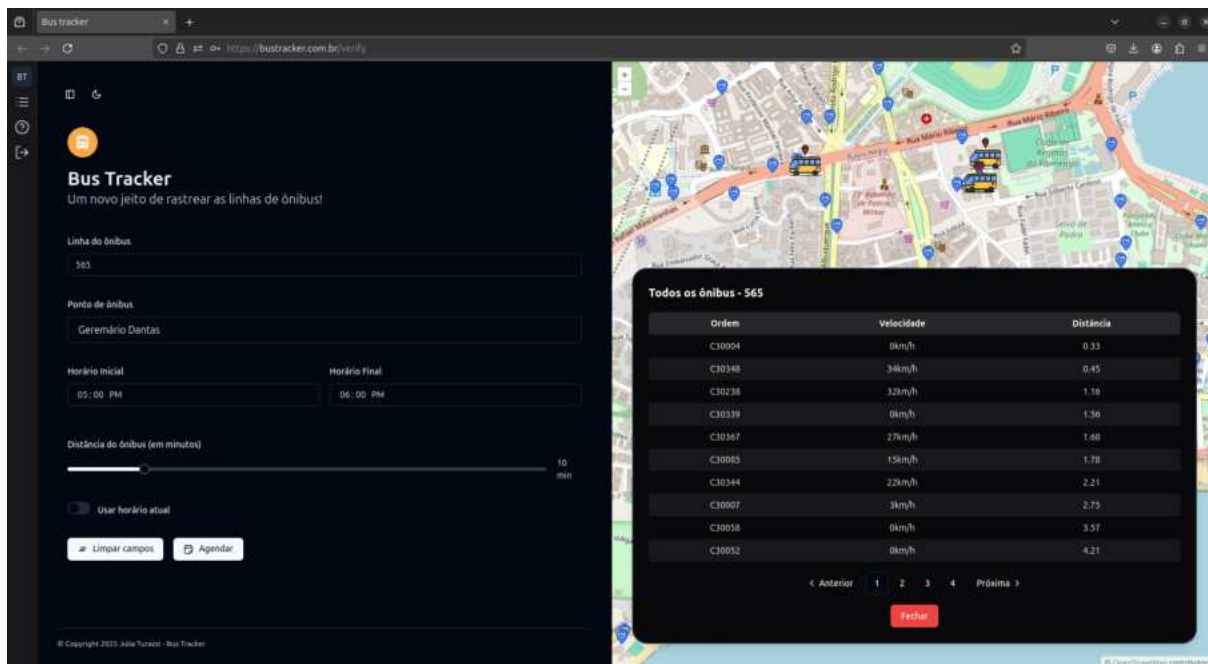


Figura 18 – Tabela das Informações dos Ônibus

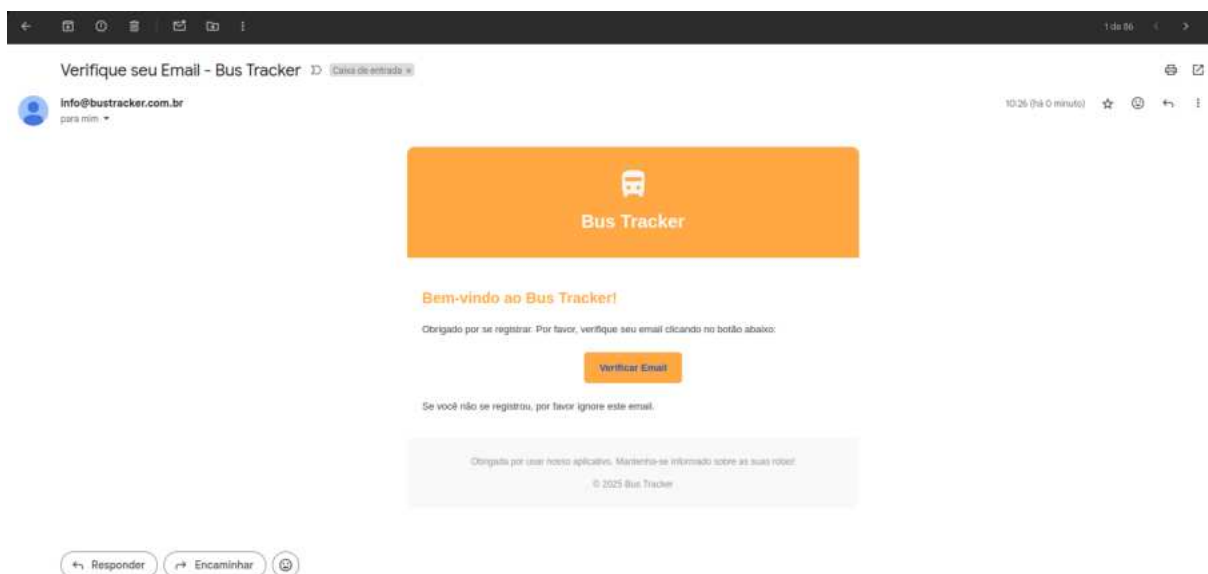


Figura 19 – Email de Verificação

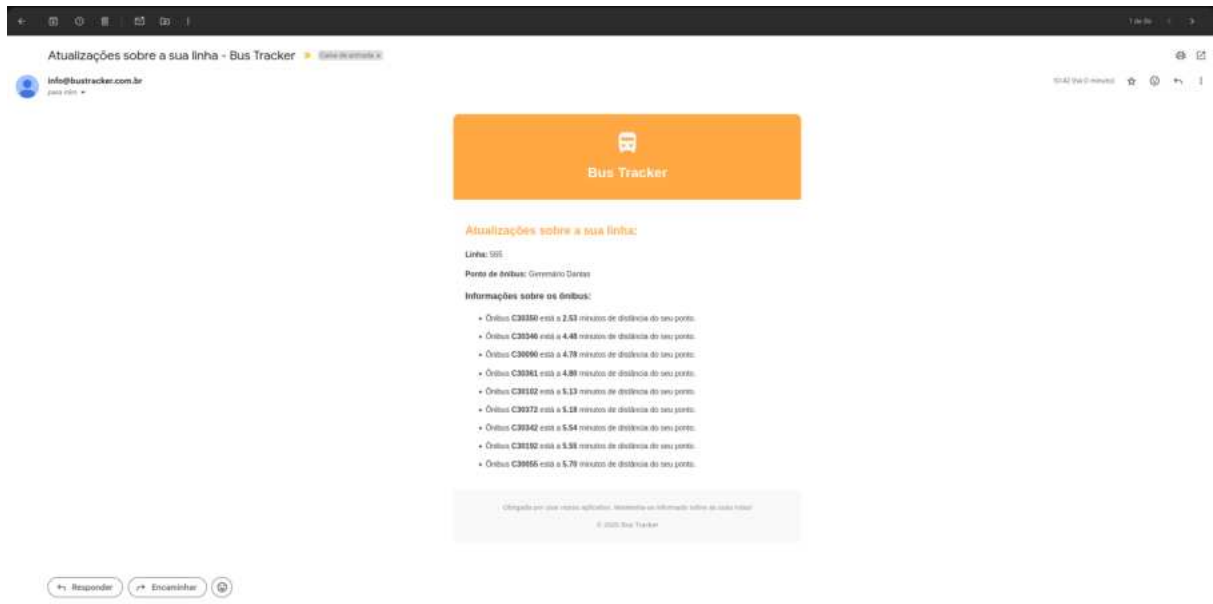


Figura 20 – Email de Atualizações

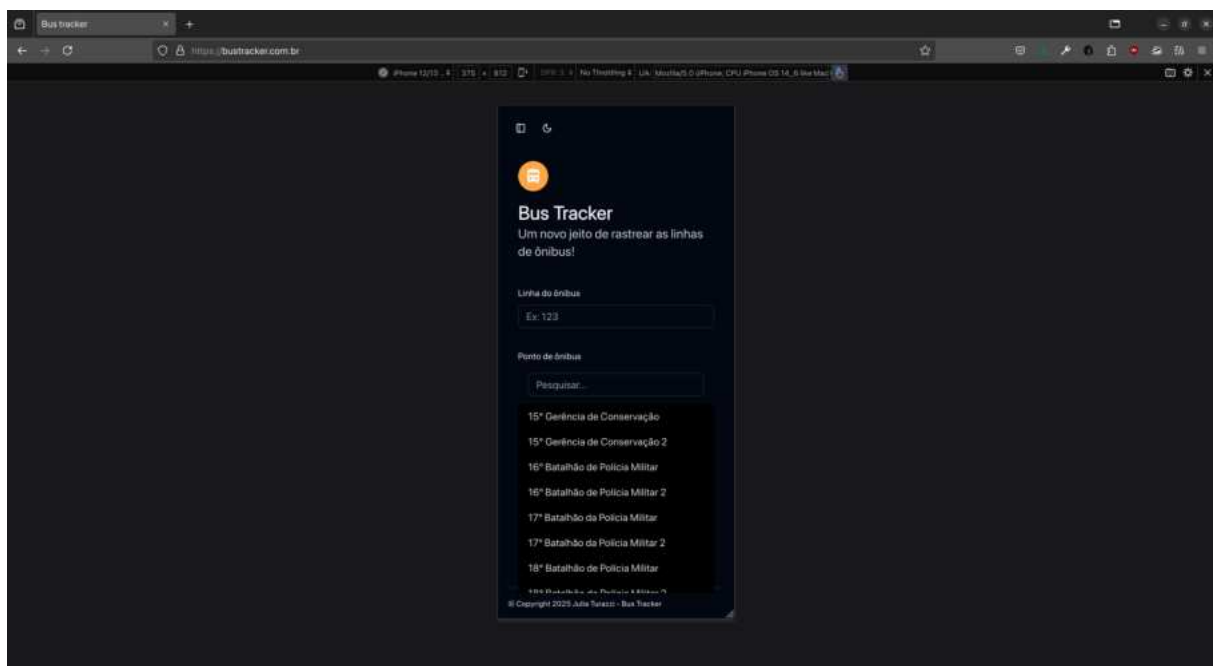


Figura 21 – Modo Mobile

APÊNDICE D – FORMULÁRIO DE AVALIAÇÃO

Avaliação do *Bus Tracker Web App*

Olá! Seja bem-vindo(a) ao Formulário de Avaliação do *Bus Tracker Web App*!

Este formulário tem como objetivo coletar suas opiniões, sugestões e experiências com o uso do nosso aplicativo durante o período de testes. Seu *feedback* é essencial para identificarmos pontos de melhoria e implementarmos novas funcionalidades que atendam melhor às suas necessidades.

Gostaríamos de ressaltar que todas as informações fornecidas são anônimas e serão utilizadas exclusivamente para fins de aprimoramento do aplicativo. Garantimos a total confidencialidade dos dados coletados.

O preenchimento deste formulário levará cerca de **5 minutos**.

Sua contribuição é fundamental para o sucesso deste trabalho. Agradecemos desde já pela sua colaboração e interesse em nos ajudar a melhorar! :)

* Indica uma pergunta obrigatória.

Você autoriza suas respostas a serem coletadas, armazenadas e analisadas? *

- Sim
- Não

Você utilizou o aplicativo como: *

- Usuário cadastrado
- Usuário não cadastrado
- Ambos

Com que frequência você usou o aplicativo durante o período de testes? *

- Apenas uma vez
- Algumas vezes
- Todos os dias

Qual dispositivo você utilizou para acessar o *Bus Tracker*? *

- Computador
- Celular
- Tablet

Como você avaliaria sua experiência geral com o *Bus Tracker*? *

- Excelente
- Boa
- Regular
- Ruim
- Muito ruim

O que mais chamou sua atenção positivamente no aplicativo? *

.....

Você conseguiu usar o aplicativo sem dificuldades? *

- Sim, foi fácil usar
- Tive algumas dificuldades, mas consegui usar
- Não, foi difícil de usar

O design e a interface do aplicativo são agradáveis? *

- Sim
- Não

O que você achou mais frustrante ou negativo no aplicativo? *

.....

A busca por pontos de ônibus funcionou como esperado? *

- Sim
- Não
- Parcialmente

O mapa em tempo atual foi útil para você? *

- Sim
- Não

Você configurou e recebeu notificações por e-mail? *

- Sim, funcionaram perfeitamente
- Sim, mas houve problemas
- Não utilizei essa funcionalidade

Há alguma funcionalidade que você gostaria de ver no aplicativo?

O aplicativo foi rápido para carregar informações? *

- Muito rápido
- Rápido
- Regular
- Lento

Você encontrou algum erro ou problema técnico durante os testes? *

Você recomendaria o *Bus Tracker* para outras pessoas?

- Sim
- Não

Você usaria o *Bus Tracker* fora do período de teste, implementando no seu dia a dia?

- Com certeza
- Provavelmente
- Talvez
- Provavelmente não
- Definitivamente não

Gostaríamos de saber sua opinião! Há mais algo que você gostaria de compartilhar sobre o *Bus Tracker*? Algum comentário ou sugestão adicional?

.....

Enviar

Página 1 de 2

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Instituto de Computação - UFRJ.
Does this form look suspicious? [Relatório](#)

Google Formulários

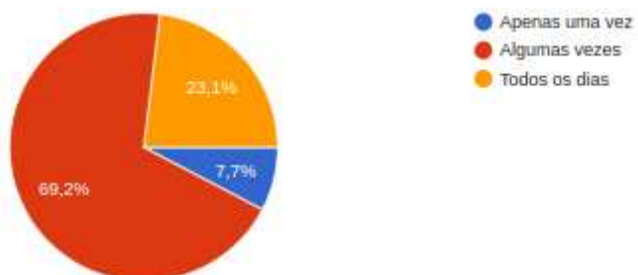
APÊNDICE E – RESULTADOS DO FORMULÁRIO DE AVALIAÇÃO



Com que frequência você usou o aplicativo durante o período de testes?

[Copiar gráfico](#)

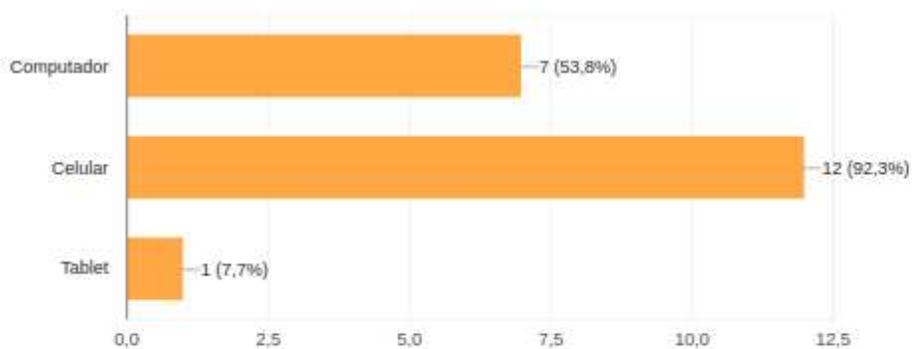
13 respostas



Qual dispositivo você utilizou para acessar o Bus Tracker?

[Copiar gráfico](#)

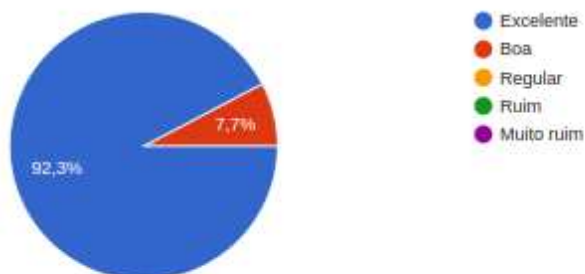
13 respostas



Como você avaliaria sua experiência geral com o Bus Tracker?

[Copiar gráfico](#)

13 respostas



| O que mais chamou sua atenção positivamente no aplicativo? |
|---|
| "A possibilidade de procurar os ônibus no mapa e em uma lista e claro, os alertas programados" |
| "Cadastro de alertas" |
| "o agendamento de alertas e a tabela de informações sobre distância em minutos (ideia genial!)" |
| "A visualização dos pontos no mapa ajudando para a escolha do mesmo" |
| "todas as funcionalidades" |
| "Os alertas de proximidade me chamaram atenção positivamente no aplicativo." |
| "O design e simplicidade desse 'web app'" |
| "As funcionalidades e o visual" |
| "o mapa mostrando os onibus e pontos" |
| "O mapa com todos os pontos e depois os ônibus" |
| "a possibilidade de receber alertas dos ônibus sem me preocupar em ficar verificando os horários ou onde eles estão." |
| "alertas em intervalos específicos de horário e com notificações; o mapa, acredito que ficou bem prático e bonito" |
| "as possibilidades de alertas e o mapa" |

Quadro 21 – Respostas sobre aspectos positivos do aplicativo

| O que você achou mais frustrante ou negativo no aplicativo? |
|--|
| "nada" |
| "Não exatamente um ponto negativo, mas acredito que para próximos passos seria legal ter informações sobre outros meios de transporte, como metrô" |
| "Nada" |
| "Adorei o aplicativo!" |
| "Nenhum em mente" |
| "Nada negativo. Apenas uma sugestão, talvez fosse interessante adicionar para as linhas uma lista de horários aproximados que normalmente os ônibus passam." |
| "Não consigo pensar em nada específico" |
| "Refletindo muito, o único ponto acredito que é a falta de possibilidade de procurar as linhas de ônibus." |
| "ser apenas do rio, o app ajudou durante a minha estadia no rio mas sou de sp" |
| "Nada, achei ótimo!" |
| "queria ter conseguido agendar por dia também, além de horário. como utilizo ônibus todo dia, gostaria de conseguir colocar um alerta todo dia no mesmo intervalo de horário." |
| "nao" |

Quadro 22 – Respostas sobre aspectos frustrantes ou negativos do aplicativo



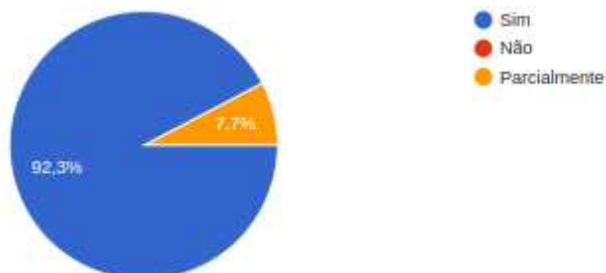
| Há alguma funcionalidade que você gostaria de ver no aplicativo? |
|---|
| "Para nova funcionalidade acredito que seria interessante adicionar a opção de agendamentos repetidos, como feito em alarmes. Ex.: Agendar apenas uma vez; Todos os dias; Dias de semana etc" |
| "a possibilidade de agendar por dia além do horário e outros modais além de ônibus (mas entendo que é a proposta desse app)" |
| "Gostaria que o mapa fosse em tempo real, atualizando com frequência depois que enviado" |
| "Não exatamente uma nova funcionalidade mas gostaria que fosse possível adicionar um intervalo maior que uma hora para os agendamentos" |
| "não" |
| "Ele ficou perfeito! Única sugestão seria, para ser mais inclusivo, talvez adicionar as informações sobre acessibilidade." |
| "Listada na pergunta: 'O que você achou mais frustrante ou negativo no aplicativo?'" |
| "nao" |

Quadro 23 – Respostas sobre funcionalidades desejadas no aplicativo

A busca por pontos de ônibus funcionou como esperado?

 Copiar gráfico

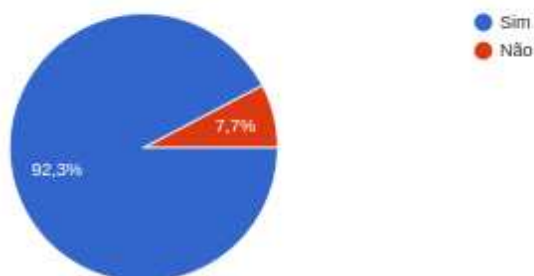
13 respostas



O mapa em tempo atual foi útil para você?

 Copiar gráfico

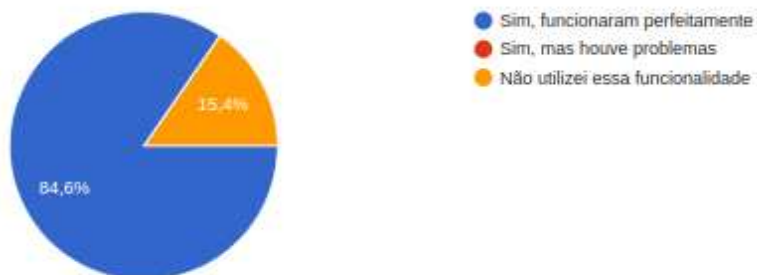
13 respostas

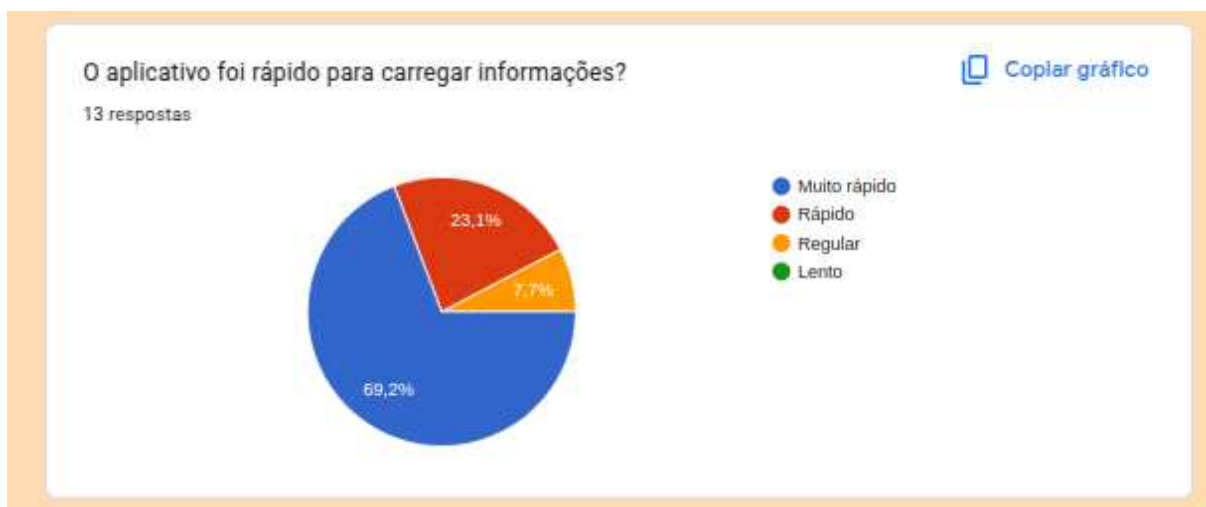


Você configurou e recebeu notificações por e-mail?

 Copiar gráfico

13 respostas





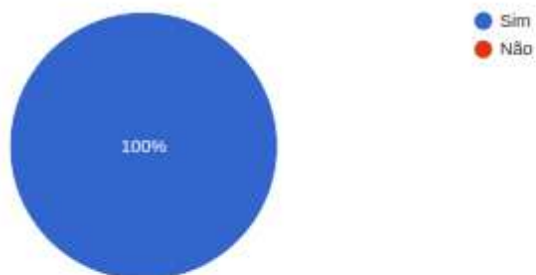
| Você encontrou algum erro ou problema técnico durante os testes? |
|---|
| "nenhum" |
| "Não" |
| "não" |
| "Nenhum problema encontrado" |
| "só uma vez que algumas distâncias nao apareceram carregadas e sim com um traço, mas logo depois, tentando novamente, funcionou normal" |
| "Não encontrei problemas técnicos" |
| "Não encontrei nenhum erro ou problema técnico durante os testes que realizei." |
| "Não me recordo de ter encontrado erros." |
| "Nenhum problema" |
| "nao" |

Quadro 24 – Respostas sobre erros ou problemas técnicos encontrados durante os testes

Você recomendaria o *Bus Tracker* para outras pessoas?

 Copiar gráfico

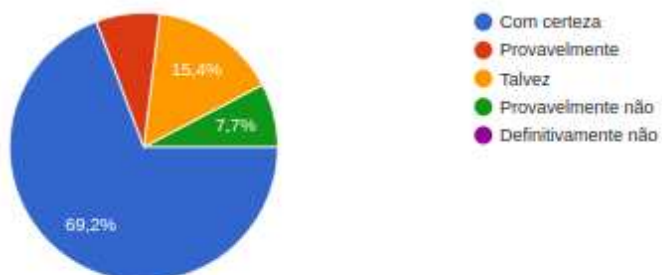
13 respostas



Você usaria o *Bus Tracker* fora do período de teste, implementando no seu dia a dia?

 Copiar gráfico

13 respostas



| |
|---|
| Gostaríamos de saber sua opinião! Há mais algo que você gostaria de compartilhar sobre o Bus Tracker? Algum comentário ou sugestão adicional? |
| "coloquei que provavelmente não usaria por ser apenas no rio, mas gostei muito desse site como um todo, bem feito e funcional" |
| "Sem comentários. As sugestões foram listadas anteriormente." |
| "Gostei bastante do aplicativo, principalmente pela facilidade de utilização, estão de parabéns pelo trabalho!" |
| "A única sugestão além da questão do mapa comentado acima, seria a possibilidade de pesquisar as linhas de ônibus, acredito que seria uma funcionalidade interessante e que poderia ajudar bastante na hora de lembrar a linha!" |
| "O trabalho ficou IMPECÁVEL, amei a oportunidade de testar o app!!" |
| "fiquei surpresa com a qualidade desse site e as funcionalidades que ele tem, parabenizo pelo esforço tanto na parte visual do site quanto nas ideias para as funcionalidades!!!" |
| "usei durante toda essa semana disponibilizada para testes e gostei bastante, deixei registrado acima alguns pontos que acredito que poderiam melhorar mas no geral achei uma ideia muito interessante e útil" |
| "Parabéns pelo trabalho, achei muito útil ;)" |
| "Acredito que esse aplicativo é uma ferramenta muito útil, com diversas funcionalidades que me ajudaram. eria incrível se aplicativos como este, especialmente o Bus Tracker, tivessem sido desenvolvidos antes, pois teriam facilitado minha vida na época em que trabalhava no Centro." |
| "Gostei demais do design e facilidade de uso desse web app, a paleta ficou ótima e principalmente a possibilidade de modificar o tema para claro ou escuro. Gostei tbm dos templates dos email e de possuir não só as instruções mas também um email para contato em caso de dúvidas" |
| "achei o aplicativo muito bem pensado e bem feito, incluindo os efeitos de algumas abas como na hora de abrir a tabela ou aquela barra lateral. ficou tudo muito bonito, incluindo os emails!" |

Quadro 25 – Respostas sobre comentários ou sugestões adicionais sobre o Bus Tracker