

OTIMIZAÇÃO DO ACESSO À INFORMAÇÃO SOBRE O TRANSPORTE PÚBLICO EM CRICIÚMA: SISTEMA PARA IDENTIFICAÇÃO DE PARADAS E ITINERÁRIOS DE ÔNIBUS

Lucas Baldasar Paquelin¹, André Faria Ruaro²

Resumo: A mobilidade urbana constitui um desafio crescente nas cidades, sobretudo devido à carência de sistemas integrados que disponibilizem informações precisas e acessíveis sobre o transporte público. Em Criciúma/SC, a ausência de uma plataforma unificada dificulta o acesso a horários, rotas e paradas. Este trabalho elaborou um aplicativo móvel que centraliza informações do transporte público municipal, promovendo maior organização, transparência e facilidade de acesso. Construído com Ionic e React, o sistema integra bibliotecas de geoprocessamento como Rlayers e Turf, Utilizando dados fornecidos pela Diretoria de Trânsito e Transporte de Criciúma. O aplicativo permite visualizar rotas e horários, identificar paradas próximas, calcular a melhor rota e estimar o tempo de chegada. Testes realizados apresentaram tempos de resposta variando entre 5 e 24 segundos, com alta aceitação por parte dos usuários, evidenciando desempenho funcional satisfatório e potencial de aplicação prática.

Palavras-chave: transporte público; geoprocessamento; mobilidade urbana; aplicativo móvel; react; ionic; openlayers; sistemas de informação geográfica; usabilidade; avaliação de desempenho.

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), lucaspaquelin@gmail.com

²Orientador. Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), andre.ruaro@unesc.net

ABSTRACT: Urban mobility is an increasingly complex challenge in cities, mainly due to the lack of integrated systems that provide accurate and accessible information about public transportation. In Criciúma/SC, the absence of a unified platform hinders access to schedules, routes, and stops. This work created a mobile application that centralizes municipal public transport information, promoting better organization, transparency, and ease of access. Built with Ionic and React, the system integrates geoprocessing libraries such as Rlayers and Turf, using data provided by the Criciúma Traffic and Transport Department. The application allows users to view routes and schedules, identify nearby stops, calculate the best route, and estimate arrival times. Tests indicated response times ranging from 5 to 24 seconds, along with high user acceptance, demonstrating functional technical performance and strong potential for practical application.

Keywords: public transport; geoprocessing; urban mobility; mobile application; react; ionic; openlayers; geographic information systems; usability; performance evaluation.

1 INTRODUÇÃO

A mobilidade urbana é um tema central ao se discutir qualidade de vida nas cidades. Sistemas de transporte público desempenham um papel crucial na redução de congestionamentos, na diminuição da emissão de poluentes e na melhoria da eficiência dos deslocamentos urbanos. Contudo, em muitas cidades, como Criciúma, a falta de tecnologias acessíveis e integradas para gerenciamento e disponibilização de informações sobre transporte público limita significativamente seu potencial de impacto. O transporte público só se torna uma opção viável se os usuários tiverem acesso fácil a informações confiáveis sobre horários e rotas (Longo, 2014).

A escolha do tema se justifica pela ausência de uma plataforma oficial que centralize as informações sobre o transporte público em Criciúma. Atualmente, os usuários precisam consultar os horários pelo site da Diretoria de Trânsito e Transporte (DTT) e recorrer a aplicativos de terceiros para obter dados sobre paradas e itinerários. Essa dispersão das informações dificulta o acesso e reduz a integração entre usuários e o sistema de transporte, evidenciando a necessidade de uma solução digital unificada. A criação de um aplicativo oficial tem o potencial de facilitar o acesso às informações, aumentar a transparência do sistema, incentivar o uso do transporte coletivo e contribuir para uma mobilidade urbana mais eficiente no município.

Neste contexto, o geoprocessamento surge como uma ferramenta estratégica. Trata-se de uma disciplina que utiliza técnicas matemáticas e computacionais para tratar informações geográficas, com aplicações em áreas como cartografia, análise de recursos naturais, energia, comunicações e planejamento urbano. Aliado aos Sistemas de Informações Geográficas (SIG), o geoprocessamento oferece soluções como mapeamento de paradas, análise de itinerários e monitoramento de fluxos de tráfego. Esses recursos possibilitam a integração de dados de diversas fontes, criando sistemas interativos que facilitam o acesso à informação e otimizam o uso dos serviços públicos (Real; Repasche; Tung, 2019).

Os STI (Sistemas de Transporte Inteligente) são sistemas que utilizam tecnologias da informação, integrando dispositivos eletrônicos, sensores e softwares para coletar, processar e distribuir informações aos usuários. O STI é uma maneira eficaz de melhorar a qualidade do transporte (Peng; Xiao; Ni, 2009).

O estudo de Lady Viviana e Leonardo Chancay (Guzmán-Palma; Chancay-García, 2023), realizado na cidade de Portoviejo, Equador, concluiu que a adoção de Sistemas Inteligentes de Transporte nos ônibus urbanos contribui para a melhoria na mobilidade e na eficiência do transporte público. Além disso, os autores destacam benefícios adicionais, como a redução de emissões e a melhoria da experiência dos usuários.

De forma semelhante, Marcello Cunha e Raissa Pereira (Rocha; Pereira, 2018) desenvolveram o aplicativo TrackBus para a cidade do Rio de Janeiro. A solução integrou dados abertos para disponibilizar informações em tempo real sobre linhas, paradas e deslocamento dos ônibus. Apesar dos desafios enfrentados com a qualidade dos dados e limitações técnicas, o trabalho demonstrou boa aceitação pelos usuários e evidenciou o potencial do uso de dados públicos e do geoprocessamento na mobilidade urbana.

Entretanto, em Criciúma, observa-se a ausência de uma plataforma oficial da Prefeitura destinada à centralização e disponibilização de informações sobre o transporte público municipal. Essa limitação dificulta o acesso dos usuários a dados completos sobre horários, itinerários e paradas, uma vez que tais informações encontram-se fragmentadas entre o site da DTT e aplicativos de terceiros. Além disso, a experiência dos usuários com essas soluções externas é frequentemente impactada pela presença de grande quantidade de anúncios, conforme relatado em comentários online (Usuários do Reddit, 2025).

O trabalho teve como ponto de partida a ausência de uma plataforma que unificasse os dados do transporte público em Criciúma. Diante dessa lacuna, o objetivo geral é desenvolver um aplicativo móvel para centralizar essas informações e otimizar a experiência do usuário. Para isso, propõe-se a criação de um sistema que reúna dados de rotas e horários em um mapa interativo e de fácil utilização.

Os objetivos específicos deste trabalho consistem em mapear e integrar os dados do transporte público de Criciúma, incluindo horários, itinerários e paradas, em uma base de dados estruturada. Além disso, desenvolver um aplicativo utilizando o framework Ionic em conjunto com a biblioteca React, assegurando compatibilidade com diferentes dispositivos móveis. Também se propõe implementar mapas interativos por meio da biblioteca Rlayers e realizar análises espaciais e operações de geoprocessamento com o uso da biblioteca Turf.

2 MATERIAIS E MÉTODOS

Esta seção detalha a arquitetura da solução proposta, descrevendo as tecnologias empregadas no desenvolvimento do aplicativo móvel, a estruturação da base de dados e os algoritmos implementados para o cálculo de rotas e estimativas de tempo.

2.1 BASE DE DADOS

A base de dados utilizada neste trabalho foi obtida diretamente junto à DTT e contém informações detalhadas sobre o transporte público da cidade. O volume de dados tabulares processados evidencia a robustez do sistema: a tabela `bus_schedules` é a mais extensa, contendo 11.977 registros de horários. A tabela `routes` armazena informações de 71 linhas distintas, enquanto a tabela `stations` possui 5 registros, referentes aos principais terminais de integração.

Complementarmente, a representação espacial utiliza arquivos no formato GeoJSON que detalham a geografia do transporte municipal. A camada referente aos itinerários (geometrias das rotas) é composta por 281 registros, enquanto a base de pontos de parada conta com 1.103 locais mapeados espalhados pela cidade. Além disso, para a contextualização geográfica, utilizou-se um arquivo contendo a delimitação de 95 bairros do município.

2.1.1 Camada de Ortofoto e Delimitação Geográfica

Para aprimorar a navegação do usuário no aplicativo, foram utilizados dados matriciais obtidos através do portal SIGSC (Sistema de Informações Geográficas de Santa Catarina), referentes à camada de Aerofotogrametria 2010 da cidade de Criciúma. Inicialmente, esses dados foram importados para o QGIS(Quantum Geographic Information System), um software livre de SIG, software de geoprocessamento, onde se procedeu à conversão das informações em uma imagem raster no formato TIFF.

Em seguida, essa imagem foi processada para a geração de raster tiles utilizando a biblioteca GDAL, por meio do comando `gdal2tiles`. O processamento foi realizado com parâmetros específicos, como `-xyz`, que define o esquema de numeração de blocos compatível com APIs modernas de mapas; `-tiledriver WEBP`, que especifica o formato de saída das imagens em WEBP, garantindo melhor compressão e desempenho; `-z 10-20`, que define os níveis de zoom gerados; e `-p mercator`, que aplica a projeção Mercator, padrão em plataformas web. Além disso, a opção `-r cubic` foi utilizada para realizar reamostragem cúbica, assegurando maior qualidade visual nas transições de escala.

Por fim, para delimitar corretamente a área de interesse, aplicou-se uma camada vetorial correspondente aos limites do município, que serviu para recortar a imagem raster, garantindo que apenas a região urbana de Criciúma fosse exibida no aplicativo.

2.2 APLICATIVO

O desenvolvimento do aplicativo foi realizado utilizando o framework Ionic (v8.0), escolhido principalmente pela disponibilidade de componentes visuais que se adaptam bem tanto a celulares quanto a tablets. Além da facilidade de criação de interfaces, o Ionic utiliza WebView para rodar a aplicação, o que foi um requisito técnico essencial para integrar a biblioteca de mapas OpenLayers (v10.4). Como essa biblioteca depende de recursos nativos de navegadores para renderizar os mapas e gerenciar interações, o ambiente do Ionic provou ser a escolha ideal para garantir o funcionamento correto das funcionalidades geográficas.

Para a construção da interface, optou-se pela utilização do React (v18.3). A escolha se justifica não apenas pela popularidade — sendo o segundo framework web mais utilizado no mundo, com 39,5% de preferência (Stack Overflow, 2024) —, mas principalmente pelo suporte que isso traz. A existência de uma comunidade ativa e de um vasto ecossistema de

bibliotecas auxiliares permitiu resolver problemas de desenvolvimento com agilidade. Essa flexibilidade do React foi fundamental para criar uma aplicação dinâmica e produtiva, facilitando a manutenção e a escalabilidade do código.

Por fim, a implementação dos mapas interativos foi feita através da biblioteca Rlayers, que adapta o OpenLayers para funcionar dentro da lógica de componentes do React, em conjunto com a ol-ext para controles de interface. Já para os cálculos matemáticos e espaciais, utilizou-se a biblioteca Turf. Essa combinação de tecnologias permitiu que o aplicativo realizasse operações complexas, como calcular distâncias e identificar rotas, de forma eficiente e diretamente no dispositivo do usuário, garantindo uma boa experiência de uso.

Para a implementação dos mapas, utilizou-se a biblioteca Rlayers, uma versão componentizada para React da OpenLayers, complementada pela biblioteca ol-ext, responsável por funcionalidades avançadas de controle e interação no mapa. Para cálculos geoespaciais mais complexos, foi empregada a biblioteca Turf, especializada em operações de análise espacial.

Para o armazenamento dos dados, foi utilizada a biblioteca sq-lite3 (v7.0.1) operando de forma nativa através do @capacitor-community/sqlite. No aplicativo, foram criadas três tabelas, a partir das tabelas fornecidas pela DTT. Foi necessário realizar algumas adaptações para compatibilidade com o SQLite, especialmente quanto aos tipos de dados. Por exemplo, campos com tipagem int(11) foram adaptados para INTEGER, pois o SQLite não possui suporte a tamanhos em inteiros. Campos definidos como varchar(255) foram adaptados para TEXT. Além disso, campos do tipo time foram substituídos por TEXT, uma vez que o SQLite não possui tipo nativo para horário.

Como pode ser observado na figura 1, a tabela stations armazena o ID de cada estação juntamente com seu nome. A tabela routes possui um ID autoincrementável, o nome da linha, o itinerário completo com todas as ruas pelas quais a linha passa, a legenda da linha, o ID da estação de partida (stations_id) e o ID da estação de chegada (stations_id1).

A tabela bus_schedules armazena o ID autoincrementável, o dia da semana em que a linha opera (diasemana), o horário de partida (horario), a legenda do horário (horario_legenda), o ID da linha correspondente (routes_id) e o ID da estação de partida (stations_id).

Figura 1 - Classes do banco de dados

```
1 CREATE TABLE IF NOT EXISTS stations (  
2   id INTEGER PRIMARY KEY,  
3   terminal TEXT NOT NULL  
4 );  
5  
6 CREATE TABLE IF NOT EXISTS routes (  
7   id INTEGER PRIMARY KEY AUTOINCREMENT,  
8   linha TEXT DEFAULT NULL,  
9   itinerario TEXT,  
10  legenda TEXT,  
11  stations_id INTEGER DEFAULT NULL,  
12  stations_id1 INTEGER DEFAULT NULL,  
13 );  
14  
15 CREATE TABLE IF NOT EXISTS bus_schedules (  
16   id INTEGER PRIMARY KEY AUTOINCREMENT,  
17   diasemana TEXT NOT NULL,  
18   horario TEXT NOT NULL,  
19   horario_legenda TEXT,  
20   routes_id INTEGER NOT NULL,  
21   stations_id INTEGER NOT NULL  
22 );
```

Fonte: Do autor.

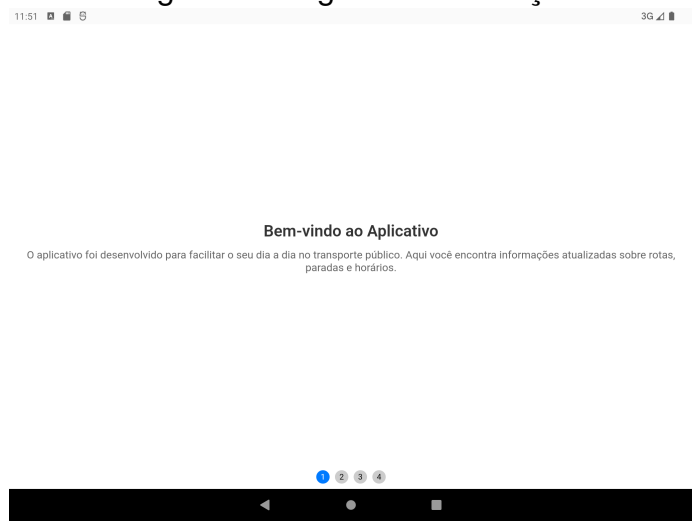
Por se tratar de um aplicativo de mapas baseado na posição do usuário, fez-se necessária a utilização de recursos de geolocalização para obter sua localização atual. Para isso, foi empregada a biblioteca `@capacitor/geolocation` (v7.0), responsável por acessar os serviços de GPS do dispositivo. A permissão de uso da localização é solicitada ao usuário tanto na tela de introdução do aplicativo quanto no momento da execução de funcionalidades que dependem diretamente dessa informação.

O gerenciamento de estado foi realizado com a biblioteca `Zustand` (v5.0.6), que se caracteriza por ser uma solução leve, simples e de alta performance. Ela permite que o estado seja chamado e manipulado a partir de qualquer parte do código, além de possibilitar a persistência de dados no armazenamento local do aplicativo.

Para a implementação das rotas, foi utilizada a biblioteca `React Router DOM` (v5.3) em conjunto com a integração do `@ionic/react-router` (v8.0.0), por meio do componente `IonReactRouter`. Essa abordagem possibilitou o gerenciamento de navegação dentro da aplicação, mantendo compatibilidade com os componentes nativos do Ionic, como `IonTabs` e `IonRouterOutlet`.

Na Figura 2, é mostrada a tela inicial do aplicativo, responsável por introduzir o sistema ao usuário. Nela, são exibidas informações de boas-vindas e instruções iniciais sobre o funcionamento do aplicativo. Em seguida, é apresentada a solicitação de permissão para acessar a localização atual, necessária para o funcionamento adequado do sistema.

Figura 2 - Página de introdução

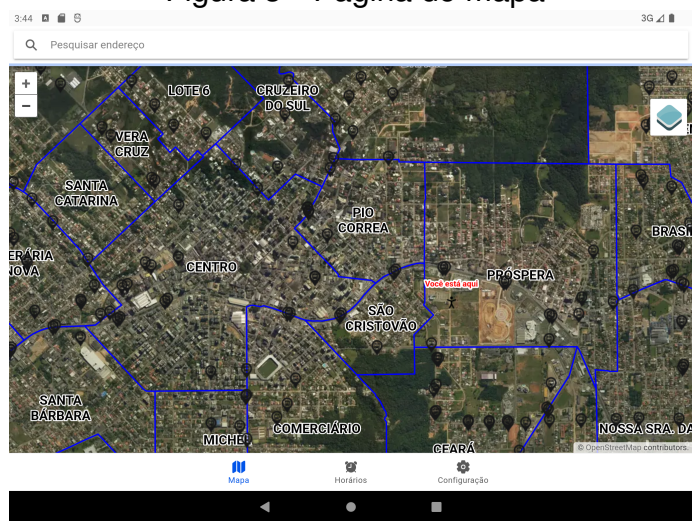


Fonte: Do autor.

A Figura 3 apresenta a tela principal do mapa do aplicativo. Nessa interface, o usuário pode visualizar o mapa com um ícone que o representa, acompanhado da mensagem “Você está aqui”, indicando sua posição atual.

Na parte superior da tela, há um campo de pesquisa destinado à inserção do nome da rua ou do local desejado. Após a inserção do destino, o aplicativo calcula automaticamente a melhor rota e ajusta o nível de zoom para exibi-la de forma adequada. O mapa apresenta o trajeto do usuário até a parada mais próxima, as paradas intermediárias, o ponto de destino e o percurso até o local final.

Figura 3 - Página do mapa



Fonte: Do autor.

No lado direito, encontra-se um ícone de menu que exibe todas as camadas disponíveis, permitindo habilitar ou desabilitar cada uma delas

e ajustar sua opacidade conforme a necessidade. Já no canto superior esquerdo, estão localizados os botões de zoom (+ e -), que possibilitam a interação direta com o mapa.

A Figura 4 apresenta a tela destinada à visualização dos horários. A interface disponibiliza um menu de seleção que lista todas as linhas, permitindo ao usuário escolher aquela de seu interesse. Uma vez definida a linha, o aplicativo exibe informações detalhadas, incluindo dias de operação, locais e horários de partida, itinerário completo e uma legenda explicativa que identifica horários especiais, como o símbolo (*) para períodos letivos, conforme os dados da base utilizada.

Figura 4 - Página dos horários



Fonte: Do autor.

2.2.1 Potencial de Expansão para Totens Interativos

A arquitetura do aplicativo foi pensada para oferecer flexibilidade e funcionar bem em diferentes tipos e tamanhos de dispositivos. O framework Ionic (v8.0) foi escolhido, entre outros motivos, por oferecer diversos componentes visuais responsivos, que se adaptam automaticamente tanto a celulares quanto a tablets.

Mesmo que o desenvolvimento até aqui tenha se concentrado em celulares e tablets, o sistema pode facilmente ser adaptado para totens públicos, permitindo que mais pessoas tenham acesso às informações do transporte coletivo em locais estratégicos, como terminais e instituições de ensino. Com isso, o aplicativo se mostra capaz de ampliar seu alcance e contribuir para uma melhor experiência de mobilidade urbana.

2.3 FUNCIONALIDADE DE BUSCA DE DESTINO

Na parte superior do aplicativo, é possível encontrar um campo que tem como objetivo permitir que o usuário digite o nome da rua ou do local de destino. Para o desenvolvimento, foi utilizada a API do LocationIQ, que é disponibilizada de forma gratuita, permitindo até 5.000 requisições por dia, duas requisições por segundo. Para utilizá-la, é necessário apenas se cadastrar na plataforma para obter a chave de uso, que é injetada na requisição. Ela possibilita realizar a requisição diretamente no navegador, sem enfrentar problemas de CORS, sendo uma ótima ferramenta, considerando sua facilidade de uso, alto desempenho e gratuidade.

Como a base de dados não é 100% completa, podem ocorrer situações em que o retorno apresente alguns problemas, como por exemplo, endereços de ruas com o mesmo nome em cidades próximas. Para contornar isso, foi implementado um filtro nos dados, validando se os campos "city" e "county" contêm o nome da cidade correta. O código foi criado para evitar possíveis inconsistências, considerando que a base de dados possui algumas limitações, as quais não prejudicam o funcionamento do sistema.

2.4 ALGORITMO DE DETERMINAÇÃO DA MELHOR ROTA

O aplicativo conta com um algoritmo desenvolvido para decidir qual a melhor rota para o respectivo cenário. O algoritmo funciona com o objetivo de escolher a melhor linha que vai passar nos próximos minutos para levar o usuário ao seu destino.

Ao selecionar o destino, o sistema busca todas as linhas que partem somente após os 10 minutos mínimos e, dentro desse conjunto, identifica qual delas sairá primeiro.

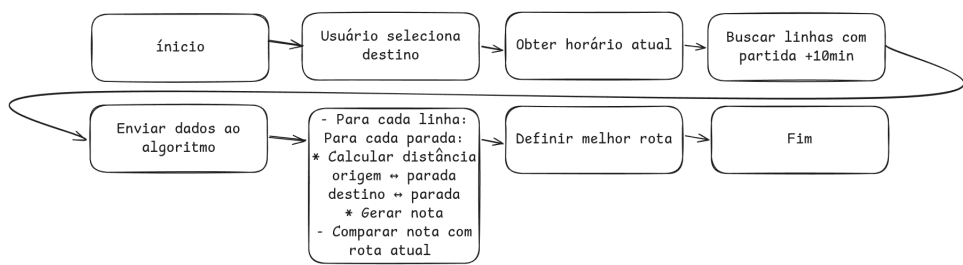
Posteriormente, são separadas as paradas que compõem cada linha. Com isso, temos as linhas e as paradas que irão partir nos próximos minutos. Após obter todos esses dados, eles são enviados para o algoritmo.

As entradas do algoritmo são: as rotas com as paradas, a coordenada de origem (localização do usuário) e a coordenada de destino. O algoritmo itera sobre as linhas, analisando uma por vez.

Após essa etapa, o algoritmo percorre as paradas da linha, calculando a distância entre cada parada e os pontos de origem e destino, utilizando a função `distance` da biblioteca `Turf`, que emprega a fórmula de Haversine para considerar a curvatura da Terra, garantindo maior precisão no cálculo das distâncias entre coordenadas geográficas (Turf.js, 2025). Dessa forma, ele obtém as menores distâncias entre o ponto de origem

e a parada de embarque, e entre a parada de destino e o ponto final. Em seguida, essas duas distâncias são somadas, e a rota com o menor valor total é definida como a melhor rota, conforme ilustrado na Figura 5.

Figura 5 - Diagrama



Fonte: Do Autor.

2.4.1 Direção da rota

Para desenvolver a lógica de identificação da melhor rota em cada caso, foi necessário também determinar o sentido de deslocamento da linha. Isso porque, na base de dados, constavam apenas o nome da linha e suas paradas associadas, sem distinção entre os sentidos de ida e volta. Como uma mesma linha pode operar em ambos os sentidos e as paradas não indicavam essa direção, foi implementada a função `nearestPointOnLine` da biblioteca `Turf.js`.

Para cada linha, os pontos correspondentes à origem e ao destino informados são projetados sobre a rota. A partir dessas projeções, torna-se possível identificar se a linha se desloca em direção ao destino ou se está retornando à origem. Assim, apenas as rotas que seguem no sentido correto, da origem ao destino, são consideradas válidas para seleção.

2.4.2 Paradas de cada linha

Para identificar as paradas correspondentes a cada linha, foi necessário aplicar um filtro nos dados, já que a única informação que indicava a quais linhas uma parada pertencia estava em um campo das propriedades do ponto no arquivo GeoJSON. Nesse campo, as linhas eram listadas da seguinte forma: "Linha: 101/404/506", sendo cada linha separada por uma barra (/). A partir dessa estrutura, foi possível filtrar e determinar a associação de cada parada às respectivas linhas. Antes disso, contudo, foi preciso padronizar os dados do campo Nome, pois não havia consistência na forma como as linhas haviam sido registradas. Por não existir um padrão no momento do salvamento das informações, muitos nomes apresentavam

diferentes caracteres para separar o nome da linha e o seu identificador, alguns utilizavam hífen, outros barra (/), e, em casos de linhas que possuíam letras, a letra podia aparecer antes ou depois do hífen. Para solucionar esse problema, realizou-se uma limpeza e padronização do campo, de modo que ele passasse a conter apenas o nome da linha.

Além desse filtro inicial, foi implementada uma lógica auxiliar para aprimorar a organização das paradas ao longo das linhas. Essa lógica atua considerando um raio de 50 metros em torno do itinerário da rota para identificar apenas as paradas que se encontram na rua por onde o ônibus vai passar e que apresentam possibilidade de parada. Em seguida, para evitar duplicidades e sobreposição de paradas de ida e volta que estejam muito próximas entre si, o algoritmo valida se não existe outra parada já selecionada a menos de 100 metros. Caso fossem mostradas e consideradas todas as paradas, isso impactaria diretamente o desempenho na escolha da melhor rota, pois haveria muito mais interações de paradas a serem processadas, além de afetar diretamente a abordagem utilizada para estimar o tempo que o ônibus levará desde sua partida até cada parada.

2.5 TEMPO ESTIMADO

Além de mostrar ao usuário a melhor rota para o determinado caso, o aplicativo também fornece uma estimativa de tempo, que vai do ponto de início da rota até a chegada à parada em que o usuário irá no ônibus. Para isso, foi considerada a rota do ônibus desde seu início até a parada de embarque, e calculada a sua distância. Após descobrir essa distância, foi adotada uma velocidade média de 60 km/h. Esse valor foi definido com base em um relatório técnico elaborado em 2021, no qual são regulamentadas as velocidades operacionais dos ônibus nas vias em que circulam, considerando faixas de 50 km/h, 60 km/h, 70 km/h e 80 km/h. Como houve maior recorrência de valores próximos a 50 km/h e 70 km/h, estabeleceu-se a média de 60 km/h como referência (Prefeitura Municipal de Criciúma, 2021).

Sabendo disso, ao dividir a distância pela velocidade média obtém-se o tempo que o ônibus levaria sem paradas. Entretanto, o ônibus realiza paradas para embarque e desembarque nas paradas ao longo do trajeto, o que demanda tempo adicional. Para isso, foi considerado 20 segundos para cada parada, do início da rota até a chegada na parada de destino, sendo esse tempo adicionado ao tempo médio. Com isso, temos o tempo estimado que o ônibus demorará para chegar à parada, que é informado ao

usuário, somado à hora de partida da linha.

2.6 MÉTODO DE AVALIAÇÃO DE USABILIDADE

Com o objetivo de avaliar a percepção dos usuários quanto à experiência de uso, foi elaborado um questionário baseado na escala de Likert, método amplamente utilizado em pesquisas de percepção e comportamento. A escala de Likert consiste em uma série de afirmações em que o participante indica seu grau de concordância em uma escala ordinal, geralmente de 1 (discordo totalmente) a 5 (concordo totalmente), permitindo quantificar opiniões subjetivas de forma estruturada e comparável (SurveyMonkey, 2025).

A avaliação da usabilidade envolveu 12 voluntários, entre usuários comuns e servidores da DTT, que testaram o aplicativo em cenários reais. Por meio de um questionário com doze questões (Apêndice A), foram analisados aspectos como facilidade de uso, clareza da interface, desempenho técnico, precisão das informações e nível de satisfação geral.

3 RESULTADOS E DISCUSSÃO

O avanço das tecnologias de geoprocessamento e das ferramentas de desenvolvimento multiplataforma tem possibilitado o desenvolvimento de aplicações voltadas à solução de problemas urbanos, especialmente no contexto da mobilidade. O sistema desenvolvido neste trabalho apresentou resultados consistentes tanto em sua execução técnica quanto em sua aplicabilidade social, demonstrando o potencial da computação aplicada à melhoria do transporte público municipal.

O aplicativo foi implementado integralmente, contemplando todas as funcionalidades propostas: visualização de rotas e horários em um mapa interativo, identificação de paradas próximas com base na localização atual do usuário, cálculo da melhor rota e estimativa de tempo de chegada.

3.1 DESEMPENHO TÉCNICO E FUNCIONAMENTO GERAL

Para a avaliação de desempenho, estabeleceu-se um protocolo de testes realizado ao longo de uma semana consecutiva, com três execuções diárias em cada período (manhã, tarde e noite). Essa mesma bateria, totalizando 63 aferições por ambiente, foi aplicada tanto no emulador (Android Studio) quanto no dispositivo físico, ambos operando com Android 12.0, visando garantir a comparabilidade dos dados. Nos testes simulados, o tempo total variou entre 6 e 15 segundos. Já no Samsung M21s, sob as

mesmas condições de sistema, o tempo médio foi maior, variando entre 20 e 24 segundos.

A etapa que mais influencia no tempo total é a organização dos dados das linhas e paradas. No Android Studio, essa etapa levou em média 4,5 segundos, enquanto no dispositivo físico apresentou cerca de 15 segundos. Essa diferença ocorre devido às limitações de processamento do aparelho em comparação ao ambiente de desenvolvimento.

Além disso, a requisição para determinar o trajeto do usuário até a parada de embarque, bem como o trajeto da parada de desembarque até o destino final, depende diretamente da qualidade da conexão com a internet do usuário, podendo aumentar a variação do tempo de resposta.

O cálculo da melhor rota em si permanece rápido, levando aproximadamente 100 milissegundos, mas o volume de dados e as condições de rede são fatores determinantes no desempenho geral.

Durante a etapa de testes, o sistema demonstrou bom desempenho na renderização das camadas de mapa e na execução das consultas ao banco de dados local (SQLite3). O uso das bibliotecas OpenLayers e Rlayers mostrou-se eficaz para o carregamento dinâmico dos itinerários e na interação com as camadas vetoriais e matriciais, incluindo a camada de ortofoto.

A funcionalidade de localização foi implementada com a biblioteca @capacitor/geolocation, permitindo ao aplicativo identificar com precisão a posição atual do usuário. Essa integração foi essencial para o funcionamento do algoritmo de determinação da melhor rota, o qual analisa as opções disponíveis em função do horário atual, distância e direção de deslocamento, retornando ao usuário o trajeto mais adequado.

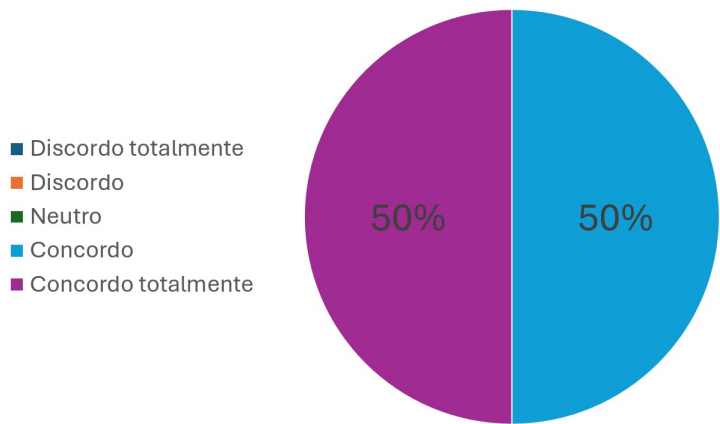
A combinação das bibliotecas Turf.js e Rlayers foi fundamental para o cálculo de distâncias e direções, garantindo que o sistema identificasse corretamente o sentido da linha (ida ou volta) e apresentasse apenas as opções viáveis.

Em relação ao design e à usabilidade, a interface foi planejada para priorizar a clareza das informações, adotando um estilo minimalista, com menus simplificados e ícones intuitivos, de modo a reduzir a poluição visual e facilitar a leitura dos elementos. O mapa interativo permite zoom, alternância de camadas e busca textual de destinos, recursos que contribuem para a fluidez da navegação e tornam o sistema acessível mesmo a usuários sem familiaridade com ferramentas de mapeamento.

3.2 AVALIAÇÃO DE USABILIDADE

A aplicação do questionário possibilitou mensurar a aceitação do sistema proposto. A Figura 6 demonstra os resultados obtidos especificamente em relação à facilidade de uso percebida pelos usuários durante os testes.

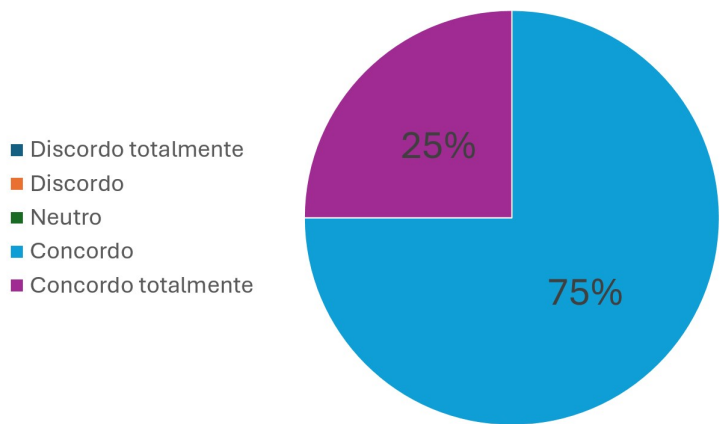
Figura 6 - Avaliação de Usabilidade do aplicativo



Fonte: Do Autor.

No que se refere à pergunta “O aplicativo é fácil de usar para encontrar rotas e paradas de ônibus.”, os dados apresentados na Figura 6 revelam que 50% dos participantes selecionaram “Concordo” e 50% “Concordo totalmente”. Essa distribuição evidencia que a interface, menus e ícones foram considerados claros e intuitivos, permitindo que os usuários naveguem pelo sistema com facilidade e encontrem rapidamente as rotas e paradas desejadas.

Figura 7 - Avaliação da Precisão das informações do aplicativo

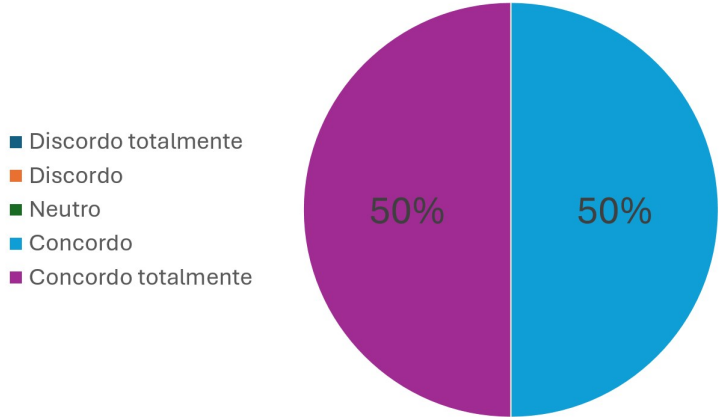


Fonte: Do Autor.

Considerando a pergunta “As rotas sugeridas pelo aplicativo são adequadas e ajudam a chegar ao destino desejado.”, observa-se, conforme

a Figura 7, que 75% dos respondentes optaram por “Concordo” e 25% por “Concordo totalmente”. Esses resultados mostram que o algoritmo de determinação da melhor rota e a identificação das paradas próximas atenderam às expectativas dos participantes, demonstrando confiança nos dados apresentados sobre horários, distâncias e trajetos.

Figura 8 - Aceitação geral do aplicativo pelos usuários



Fonte: Do Autor.

Em relação à pergunta “Você recomendaria o uso deste aplicativo para outras pessoas?”, associada à Figura 8, verifica-se que 50% dos participantes indicaram “Concordo” e 50% “Concordo totalmente”. Isso confirma que o aplicativo é percebido como uma ferramenta útil e relevante para o planejamento de deslocamentos diários, contribuindo para um transporte público mais acessível e eficiente.

De forma geral, as respostas do questionário revelaram uma percepção amplamente positiva entre os participantes, com 58,3% das respostas indicando “Concordo”, 37,5% “Concordo totalmente” e apenas 4,2% neutras, sem registros de discordância. Essa predominância de avaliações favoráveis demonstra a boa aceitação do aplicativo e a satisfação dos usuários quanto à sua funcionalidade. Os resultados reforçam que o sistema atendeu às expectativas em termos de usabilidade, clareza e utilidade prática, contribuindo de forma significativa para a experiência de uso e para o acesso a informações sobre o transporte público de Criciúma.

As questões referentes à adequação das rotas sugeridas e à confiabilidade do tempo estimado de chegada obtiveram resultados consistentes, demonstrando que o algoritmo de determinação da melhor rota atendeu às expectativas quanto à precisão e funcionalidade. Da mesma forma, os itens voltados ao desempenho geral e à estabilidade do aplicativo mostraram alto índice de concordância, o que indica que o sistema

operou sem travamentos e com bom tempo de resposta durante os testes.

Os aspectos visuais também foram bem avaliados: o design do mapa e a visualização das rotas e paradas receberam respostas predominantemente positivas. Além disso, as perguntas sobre utilidade prática e apoio ao planejamento de deslocamentos evidenciaram o potencial do sistema em auxiliar usuários do transporte público na organização de suas viagens diárias.

Quanto aos aspectos visuais, o design do mapa e a visualização das rotas e paradas foram considerados claros e de fácil interpretação, o que reforça a eficiência das escolhas gráficas e da hierarquia de informações adotadas. As questões sobre utilidade prática e apoio ao planejamento de deslocamentos mostraram que o aplicativo tem potencial real de auxiliar a população no uso do transporte coletivo, ao reunir em um único ambiente dados antes dispersos.

Por fim, as questões sobre satisfação geral e intenção de recomendação obtiveram as maiores médias do questionário, evidenciando a boa aceitação do sistema e reconhecendo sua relevância como ferramenta de apoio à mobilidade urbana. Esses resultados confirmam que o aplicativo cumpre seu propósito central de facilitar o acesso a informações sobre linhas, horários e paradas, contribuindo para um transporte público mais acessível e eficiente na cidade de Criciúma.

3.3 IMPACTO SOCIAL E APLICABILIDADE

Ao final do desenvolvimento do aplicativo, foi realizada uma reunião com representantes da DTT de Criciúma, com o objetivo de apresentar a solução desenvolvida e discutir seu potencial de aplicação no contexto municipal. Durante o encontro, os gestores demonstraram apreço pela iniciativa e pelo resultado apresentado, destacando a clareza na visualização das linhas, itinerários e horários.

Ressalta-se que o aplicativo foi desenvolvido a partir de um banco de dados reduzido fornecido pela DTT, contendo informações essenciais, mas não completas. Apesar dessa limitação, o sistema foi capaz de processar e exibir informações relevantes de forma funcional, o que evidencia a robustez do algoritmo de determinação de rotas e a eficiência da interface interativa.

Do ponto de vista social, o aplicativo contribui diretamente para a melhoria do acesso à informação sobre o transporte público, centralizando dados que antes estavam dispersos e facilitando o planejamento das via-

gens diárias dos usuários.

Em relação à escalabilidade, a arquitetura do sistema permite a expansão da base de dados e a inclusão de novas funcionalidades, como acompanhamento em tempo real dos veículos. Além disso, é viável integrar o aplicativo a APIs externas para consulta online das informações, eliminando a necessidade de armazenar os dados diretamente no aplicativo e permitindo que as atualizações sejam refletidas em tempo real. Essa abordagem amplia o potencial de replicação da solução em diferentes cidades, reforçando a capacidade do sistema de se adaptar a contextos urbanos variados e à evolução das fontes de dados disponíveis.

3.4 LIMITAÇÕES DO SISTEMA

O sistema apresenta algumas limitações decorrentes de suas dependências técnicas e de projeto. A principal delas está relacionada à necessidade de conexão com a internet para o funcionamento de determinadas funcionalidades, como a busca por destinos. Essa operação depende de uma API externa responsável por retornar os resultados conforme o texto digitado pelo usuário, o que impossibilita o uso completo do aplicativo em locais sem acesso à rede.

Outra limitação refere-se à disponibilidade do aplicativo apenas para o sistema operacional Android. Apesar de o código desenvolvido ser multiplataforma, a geração da versão para iOS requer um computador com o sistema macOS, necessário para realizar o processo de compilação e assinatura do aplicativo voltado à App Store. Assim, a limitação não está na arquitetura do sistema, mas sim na ausência do ambiente adequado para a compilação da versão iOS.

Também se destaca a ausência de dados em tempo real sobre a localização dos ônibus. Caso tais informações estivessem disponíveis e integradas ao sistema, seria possível oferecer maior precisão quanto ao tempo estimado de chegada dos veículos às paradas, aprimorando significativamente a experiência dos usuários.

Outra limitação técnica está relacionada ao uso do serviço LocationIQ para a busca de endereços e locais. Essa API impõe um limite de até 5.000 requisições diárias e permite apenas duas requisições por segundo, o que foi suficiente durante o processo de desenvolvimento e testes. No entanto, para uma versão em produção e de uso público, seria necessário adotar um serviço mais robusto, com suporte a um volume maior de requisições diárias e simultâneas, além de uma base de dados mais consolidada

para garantir maior precisão e abrangência nas buscas por ruas e pontos de interesse.

Por fim, destaca-se que as camadas cartográficas, como a camada de ortofoto, estão atualmente armazenadas dentro do próprio aplicativo. Essa abordagem permite o carregamento do mapa mesmo sem conexão com a internet, oferecendo maior autonomia ao usuário. Contudo, essa estratégia contribui para o aumento do tamanho final do aplicativo, tornando-o mais pesado e ocupando mais espaço de armazenamento nos dispositivos. Em uma versão voltada à produção, as camadas seriam disponibilizadas em um banco de dados remoto e obtidas por meio de requisições controladas, o que reduziria o tamanho do aplicativo e facilitaria futuras atualizações e manutenção das imagens cartográficas.

3.5 COMPARAÇÃO COM TRABALHOS CORRELATOS

Um trabalho que apresenta proposta semelhante é o TrackBus, (Rocha; Pereira, 2018), que apresentou um aplicativo para monitorar em tempo real os ônibus do Rio de Janeiro, utilizando dados abertos do portal Data.Rio. Assim como o sistema proposto para Criciúma, o TrackBus buscou facilitar o acesso às informações de transporte público, exibindo no mapa as linhas, paradas e notificações de proximidade. A principal diferença entre os projetos está na origem e tratamento dos dados.

Enquanto o TrackBus se baseou em informações disponibilizadas publicamente pela prefeitura do Rio de Janeiro, este trabalho utilizou uma base estruturada em parceria direta com a DTT, permitindo maior controle e consistência sobre os dados.

Uma diferença funcional ainda mais significativa é a capacidade de planejamento de viagens. Conforme relatado pelos autores do TrackBus, o aplicativo não incluiu um algoritmo que sugere a melhor rota com base na origem e destino, sendo esta uma decisão de projeto para focar em outras funcionalidades, como o monitoramento em tempo real. O presente trabalho, em contrapartida, implementa como funcionalidade central o algoritmo de determinação da melhor rota (detalhado na Seção 2.4). Essa abordagem preenche a lacuna de planejamento ativo, calculando a jornada ideal para o usuário com base no horário e na minimização de distâncias a pé, em vez de focar apenas na consulta passiva de linhas.

O presente trabalho se diferencia por ser tecnologicamente mais moderno, com design enxuto, foco na usabilidade e integração direta com dados municipais.

4 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um aplicativo móvel cujo objetivo central foi otimizar o acesso à informação sobre o transporte público de Criciúma, reunindo em uma única plataforma dados de rotas, horários e paradas fornecidos pela DTT. A solução implementada, baseada nas tecnologias Ionic e React e apoiada pelas bibliotecas Rlayers, OpenLayers e Turf.js, demonstrou ser viável para mapear itinerários, identificar paradas próximas e calcular uma estimativa de tempo de chegada, cumprindo os objetivos propostos e oferecendo uma interface intuitiva para o usuário final.

Os resultados obtidos nas etapas de teste e avaliação indicam desempenho técnico eficiente: o carregamento das camadas cartográficas e as consultas ao banco SQLite apresentaram boa responsividade, e o algoritmo de determinação da melhor rota mostrou-se consistente na seleção de trechos e paradas mais adequados. A validação com usuários, realizada com 12 voluntários, demonstrou predominância de respostas favoráveis, evidenciando aceitação quanto à clareza da interface, utilidade prática e confiabilidade das rotas e estimativas fornecidas. Além disso, a apresentação do protótipo à DTT recebeu retorno favorável, indicando potencial de aplicabilidade municipal.

Apesar dos resultados promissores, o sistema ainda possui limitações relevantes que orientam sua evolução: a versão atual foi validada principalmente em Android; a busca por destinos depende de um serviço externo com limite diário de requisições; não há integração com dados de localização dos veículos em tempo real; e parte das camadas cartográficas está embutida no aplicativo, aumentando seu tamanho. Essas restrições não comprometem a proposta central, mas apontam para melhorias necessárias antes de uma adoção em larga escala.

Trabalhos futuros incluem: geração e validação da compilação para iOS; integração de dados em tempo real sobre a localização dos veículos; implementação de apresentação de múltiplas opções de rota com critérios de ordenação (tempo, distância a pé, etc.); otimização do tamanho do aplicativo por meio de tiles remotos, cache e compressão; e externalização de cálculos complexos para uma API centralizada, reduzindo processamento local. Essas direções visam tornar o sistema mais escalável, preciso e adequado a ambientes de produção, ampliando seu impacto na mobilidade urbana e facilitando sua replicação em outros municípios.

REFERÊNCIAS

GUZMÁN-PALMA, L. V.; CHANCAY-GARCÍA, L. **Estudio bibliográfico de sistemas de transporte inteligente orientado a los buses urbanos de la ciudad portoviejo**. Polo del Conocimiento: Revista Científico-Profesional, 2023, v. 8, n. 5, p. 1200–1218. Acesso em: 19 mai. 2025. Disponível em: <<https://dialnet.unirioja.es/servlet/articulo?codigo=9335831>>.

LONGO, M. R. **O transporte sobre trilhos como melhoria para a mobilidade urbana na região metropolitana de são paulo**. São Paulo, 2014. Acesso em: 14 mar. 2025. Disponível em: <<https://dx.doi.org/10.5821/siiu.6212>>.

PENG, L.; XIAO, F.; NI, Z. Design for wireless sensor network-based intelligent public transportation system. In: **2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication**. [S.l.: s.n.], 2009. p. 351–354.

Prefeitura Municipal de Criciúma. **Levantamento Técnico: Controlador de Velocidade**. Criciúma, SC, 2021. Referente ao biênio 2021/2023, conforme Resolução 798/2020 do CONTRAN.

REAL, V. K. C.; REPASCHE, K. A.; TUNG, L. N. Infobus – projeto instrucional sobre mobilidade urbana e transporte público na cidade de campinas/sp. In: **9º Congresso Internacional de Design da Informação**. [s.n.], 2019. Acesso em: 22 abr. 2025. Disponível em: <<https://www.proceedings.blucher.com.br/article-details/infobus-projeto-instrucional-sobre-mobilidade-urbana-e-transporte-pblico-na-cidade-c>>

ROCHA, M. C. d.; PEREIRA, R. **Trackbus: um aplicativo de apoio aos usuários de ônibus na cidade do Rio de Janeiro**. 2018. Acesso em: 3 out. 2025. Disponível em: <<https://app.uff.br/riuff/handle/1/5807>>.

Stack Overflow. **Stack Overflow Developer Survey 2024: Most Popular Web Frameworks**. 2024. Acesso em: 24 mai. 2025. Disponível em: <<https://survey.stackoverflow.co/2024/technology#most-popular-technologies-webframe>>.

SurveyMonkey. **Escalas de Likert: exemplos, dicas e como analisar os dados**. 2025. Acesso em: 21 out. 2025. Disponível em: <<https://pt.surveymonkey.com/mp/likert-scale>>.

Turf.js. **Distance – Turf.js API Documentation**. 2025. Acesso em: 2 nov. 2025. Disponível em: <<https://turfjs.org/docs/api/distance>>.

Usuários do Reddit. **Qual aplicativo de ônibus você usam?** 2025. Acesso em: 21 out. 2025. Disponível em: <https://www.reddit.com/r/saopaulo/comments/1jj5ya3/qual_aplicativo_de_onibus_vc_usam_j%C3%A1_que_o_moovit/>.

A APÊNDICE - Enunciado das Perguntas do Formulário de Avaliação

Este formulário foi utilizado para coletar as opiniões dos usuários que embasam os resultados apresentados na Seção 3.

1. O aplicativo é fácil de usar para encontrar rotas e paradas de ônibus?
2. A interface do aplicativo (menus, botões, ícones e mapa) é clara e intuitiva?
3. O aplicativo identifica corretamente minha localização e mostra as paradas de ônibus próximas?
4. As rotas sugeridas pelo aplicativo são adequadas e ajudam a chegar ao destino desejado?
5. O tempo estimado de chegada e as informações de trajeto são confiáveis?
6. O aplicativo é rápido e funciona de forma estável (sem travamentos ou erros)?
7. O design do mapa e a visualização das rotas e paradas são claros e fáceis de entender?
8. O aplicativo ajuda a planejar melhor o deslocamento no transporte público?
9. O desempenho do aplicativo (velocidade, carregamento do mapa) atendeu às suas expectativas?
10. O aplicativo seria útil no seu dia a dia para se deslocar pela cidade?
11. Qual o seu nível geral de satisfação com o aplicativo?
12. Qual era o destino que você queria alcançar?