



Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas
Colegiado de Engenharia de Computação**

Felipe Afonso Rezende Vieira

**UM SISTEMA PARA
ACOMPANHAMENTO DA
LOCALIZAÇÃO EM TEMPO REAL
DE VEÍCULOS DO TRANSPORTE
PÚBLICO**

Julho, 2018

João Monlevade—MG

Felipe Afonso Rezende Vieira

UM SISTEMA PARA ACOMPANHAMENTO DA LOCALIZAÇÃO EM TEMPO REAL DE VEÍCULOS DO TRANSPORTE PÚBLICO

Orientador: George Henrique Godim da Fonseca

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Julho de 2018

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

UM SISTEMA PARA ACOMPANHAMENTO DA LOCALIZAÇÃO EM TEMPO REAL DE VEÍCULOS DO TRANSPORTE PÚBLICO

Felipe Afonso Rezende Vieira

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação e aprovada pela Banca Examinadora abaixo assinada:



George Henrique Godim da Fonseca
Doutor em Engenharia Elétrica
DECSI – UFOP



Gilda Aparecida de Assis
Doutora em Engenharia Elétrica
Examinador
DECSI – UFOP



Harlei Miguel de Arruda Leite
Doutor em Engenharia Elétrica
Examinador
DECSI – UFOP

João Monlevade, 19 de julho de 2018

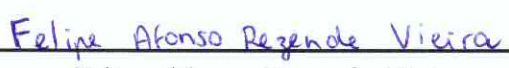
ATA DE DEFESA

No dia 19 do mês de Julho de 2018, às 18:00 horas, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Felipe Afonso Rezende Vieira**, sendo a Comissão Examinadora constituída pelos professores: George Henrique Godim da Fonseca, Gilda Aparecida de Assis, Harlei Miguel de Arruda Leite. O(a) candidato(a) apresentou a monografia intitulada: **“UM SISTEMA PARA ACOMPANHAMENTO DA LOCALIZAÇÃO EM TEMPO REAL DE VEÍCULOS DO TRANSPORTE PÚBLICO”**. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 10.0 (dez pontos), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).


George Henrique Godim da Fonseca
Doutor em Engenharia Elétrica
DECSI – UFOP


Gilda Aparecida de Assis
Doutora em Engenharia Elétrica
Examinador(a)
DECSI – UFOP


Harlei Miguel de Arruda Leite
Doutor em Engenharia Elétrica
Examinador(a)
DECSI – UFOP


Felipe Afonso Rezende Vieira

João Monlevade, 19 de julho de 2018

TERMO DE RESPONSABILIDADE

Eu, **Felipe Afonso Rezende Vieira** declaro que o texto do trabalho de conclusão de curso intitulado “**UM SISTEMA PARA ACOMPANHAMENTO DA LOCALIZAÇÃO EM TEMPO REAL DE VEÍCULOS DO TRANSPORTE PÚBLICO**” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 19 de julho de 2018

Felipe Afonso Rezende Vieira

Felipe Afonso Rezende Vieira

*Dedicado aos meus pais e minha irmã, cuja compreensão e apoio tornaram possível a
realização desse trabalho.*

Agradecimentos

A Deus, por ter me dado saúde, força e disposição para superar as dificuldades e acordar diariamente motivado a me empenhar em minhas metas e projetos pessoais.

A minha família, especialmente ao meu pai Mário Afonso Vieira, minha mãe Luzia Aparecida de Rezende e a minha irmã Patrícia Cristina Rezende Vieira, por sempre acreditar e torcer por mim, me incentivar e compreender minha ausência diversas vezes durante o período da graduação.

Ao meu orientador, pela paciência, disposição e apoio proporcionado durante todo o trabalho.

Aos meus amigos e colegas de curso, pelo companheirismo e pela ajuda fornecida em todos os momentos que precisei.

A todos os professores que de fato se dedicam a nobre missão de transmitir o conhecimento, posso afirmar, sem dúvidas, que contribuíram muito para minha formação pessoal e profissional.

“Eu não tenho ídolos. Tenho admiração por trabalho, dedicação e competência.”

— Ayrton Senna

Resumo

Este trabalho apresenta o desenvolvimento de um sistema para acompanhamento em tempo real dos ônibus do transporte público urbano, inicialmente voltado à cidade de João Monlevade/MG. O objetivo do trabalho consiste em apresentar uma solução para minimizar o tempo de espera em paradas de ônibus, propondo uma arquitetura que não necessite de *hardwares* específicos e que seja funcionalmente e economicamente viável.

Foram desenvolvidos dois aplicativos para dispositivos móveis que utilizam o sistema operacional Android, sendo um responsável pela coleta e envio da localização dos ônibus para um servidor e outro que possibilita aos usuários o acompanhamento dos itinerários, localização e horário dos ônibus. Além disso, foi feita a implementação do *web service* em Java que permite o acesso e manipulação do banco de dados. Os resultados obtidos foram satisfatórios e demonstraram que é possível fornecer a localização dos ônibus com alto grau de confiabilidade, precisão e com baixo consumo de dados móveis.

Palavras-chaves: Localização. Tempo real. Transporte coletivo. Android.

Abstract

This document presents the development of a system to monitor in real time the location of urban transport vehicles, initially aimed at the city of João Monlevade/MG. The purpose of this document consists in presenting a solution to the waiting time in the bus stops, proposing an architecture that does not require specific hardware and that is functionally and economically feasible. Two Android mobile that use the Android operating system were developed, one being responsible for collecting and sending the location of the buses to a server and another that allow users to see the itineraries, location and time of the bus. In addition, an implementation of the Java web service was made that allows access and manipulation of the database. The results obtained were satisfactory and demonstrated that it is possible to provide a real time bus locations with high reliability, precision and low data usage.

Key-words: Location. Real-time. Collective transport. Android.

Lista de ilustrações

Figura 1 – Visão geral da arquitetura do sistema	21
Figura 2 – Diagrama de Casos de Uso	24
Figura 3 – Diagrama de Atividades do aplicativo do motorista	28
Figura 4 – Diagrama de Atividades do aplicativo dos passageiros	29
Figura 5 – Diagrama de Classes do aplicativo do motorista	30
Figura 6 – Diagrama de Classes do aplicativo dos passageiros	32
Figura 7 – Tela inicial do aplicativo do motorista	35
Figura 8 – Conexões desabilitadas	36
Figura 9 – Alerta para necessidade de seleção do número e da linha do ônibus	36
Figura 10 – Mapa com a rota e posição atual do ônibus	37
Figura 11 – Notificação com <i>status</i>	37
Figura 12 – Confirmação de término do percurso	37
Figura 13 – Caixa de diálogo para definição do intervalo entre envios	38
Figura 14 – <i>Splash Screen</i>	39
Figura 15 – Mapa indicando a posição do usuário	39
Figura 16 – Tela com lista de linhas para acompanhamento	39
Figura 17 – Mensagens informativas	40
Figura 18 – Rota e localizações exibidas após seleção	40
Figura 19 – Duas linhas sendo exibidas simultaneamente	41
Figura 20 – Confirmação para limpar o mapa	41
Figura 21 – Caixa de diálogo para seleção do intervalo de atualização desejado	42
Figura 22 – Lista de linhas de ônibus	42
Figura 23 – Quadro de horários de partida	43
Figura 24 – Exibição das informações complementares	43

Lista de tabelas

Tabela 1 – Comparativo com os sistemas relacionados	19
---	----

Lista de quadros

Quadro 1 – Caso de Uso - Iniciar monitoramento	25
Quadro 2 – Caso de Uso - Finalizar monitoramento	26
Quadro 3 – Caso de Uso - Visualizar ônibus e itinerário	26
Quadro 4 – Caso de Uso - Visualizar horários	27
Quadro 5 – Caso de Uso - Configurar período entre atualizações	27

Lista de abreviaturas e siglas

FGV	Fundação Getúlio Vargas
GB	Gigabyte
GPRS	Global Positioning System
IDE	Integrated Development Environment
iOS	iPhone Operating System
MG	Minas Gerais
KML	Keyhole Markup Language
RAM	Random Access Memory
RFID	Radio-Frequency IDentification
SGBD	Sistema de Gerenciamento de Banco de Dados
SIT	Sistemas Inteligentes de Transporte
SOAP	Simple Object Access Protocol
SP3	Service Pack 3
SPTrans	São Paulo Transportes
SQL	Structured Query Language
UML	Unified Modeling Language
WAR	Web Application Archive
XML	Extensible Markup Language

Sumário

1	INTRODUÇÃO	15
1.1	Problema	15
1.2	Objetivos	16
1.3	Justificativa	16
1.4	Estrutura do trabalho	16
2	CONCEITOS GERAIS E REVISÃO DA LITERATURA	17
3	METODOLOGIA	20
3.1	Visão geral do sistema	20
3.2	Ferramentas e tecnologias utilizadas	21
3.3	Especificações	22
3.3.1	Requisitos funcionais	22
3.3.2	Requisitos não funcionais	23
3.4	Diagrama de Casos de Uso	23
3.5	Diagramas de Atividades	27
3.6	Diagramas de Classes	30
3.7	Banco de Dados	33
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	34
4.1	Aplicativo dos motoristas	34
4.2	Aplicativo dos passageiros	38
4.3	Testes realizados	44
5	CONCLUSÃO	45
	REFERÊNCIAS	47

1 Introdução

A falta de informações precisas com relação à localização e ao horário que os ônibus chegam aos pontos pode fazer com que os usuários do transporte público fiquem durante longos períodos esperando sem ter nem sequer alguma estimativa de horário. Muitas vezes os usuários precisam aguardar em locais sem estrutura e com pouca segurança, estando suscetíveis a assaltos. Uma forma de resolver essa situação é através do monitoramento da posição geográfica dos ônibus, utilizando sensores GPS. Dessa forma, as coordenadas geográficas obtidas podem ser disponibilizadas aos usuários para que eles possam acompanhar e planejar melhor a ida para o ponto, visando diminuir ao máximo o tempo de espera. O propósito desse trabalho consiste na arquitetura e desenvolvimento de um sistema para o monitoramento em tempo real dos ônibus do transporte público, inicialmente voltado à cidade de João Monlevade/MG.

O surgimento dos *smartphones* proporcionou muita comodidade às pessoas: novas formas de contato social foram desenvolvidas, aplicativos de entretenimento, fotografia, compras, música e comunicação se popularizaram e estão disponíveis em lojas virtuais por onde podem ser rapidamente baixados e instalados. Em 2018, o Brasil superou a marca de um *smartphone* por habitante e conta com 220 milhões de celulares inteligentes ativos, de acordo com a 29ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela FGV ([MEIRELLES, 2018](#)).

Considerando a popularização dos *smartphones* e sua infinidade de recursos, pode-se inferir que soluções que visam resolver problemas que envolvem a sociedade de uma forma geral, devem oferecer formas de acesso simples, difundidas e que estejam ao alcance do usuário de forma imediata. Utilizando-se apenas *smartphones* é possível desenvolver uma solução para monitorar a localização dos ônibus e fornecer ao usuário essa informação em tempo real, permitindo uma melhora na qualidade do transporte oferecido e evitando que o usuário fique por muito tempo aguardando.

1.1 Problema

Atualmente, o sistema de transporte público, na grande maioria das cidades brasileiras, encontra-se em situação caótica. A extensa lista de problemas que frequentemente são encontrados deixou de ser uma realidade apenas dos grandes centros urbanos e já faz parte da realidade de cidades de médio e pequeno porte.

Dentre os problemas verificados com maior constância, encontram-se os preços que não condizem com a qualidade do transporte oferecido, ônibus frequentemente cheios,

veículos em condições ruins e o grande tempo de espera nos pontos de ônibus.

Esse trabalho tem como finalidade minimizar um desses problemas, visando diminuir o tempo de espera nas paradas de ônibus através da utilização de uma aplicação que possibilite o monitoramento, em tempo real, da localização dos ônibus, permitindo que os usuários possam planejar melhor a saída de casa e do trabalho.

1.2 Objetivos

O principal objetivo desse trabalho consiste em arquitetar e desenvolver um sistema que possibilite aos usuários do transporte público acompanhar em tempo real a localização dos ônibus. Pretende-se desenvolver dois aplicativos para dispositivos móveis que utilizam o sistema operacional Android, sendo um responsável pela coleta e envio da localização dos ônibus para um servidor e outro que possibilite aos usuários o acompanhamento dos itinerários, pontos de parada e a localização dos ônibus.

Como objetivos específicos do presente trabalho podem-se citar:

- Permitir aos usuários acompanhar a localização dos ônibus em tempo real;
- Fornecer os horários de partida dos ônibus;
- Disponibilizar no mapa o trajeto/rota a ser percorrido pelos ônibus.

1.3 Justificativa

A elaboração do trabalho se justifica no grande benefício que pode ser proporcionado a todos os usuários do transporte público, principalmente na cidade de João Monlevade, onde não existe nenhum aplicativo semelhante que ofereça esse serviço.

1.4 Estrutura do trabalho

No presente capítulo é feita a contextualização do tema abordado e são expostos os objetivos da proposta de trabalho, bem como sua justificativa. No [Capítulo 2](#), são expostas algumas aplicações da tecnologia nos sistemas de transporte, e são destacados os principais aplicativos existentes com funcionalidades semelhantes. No [Capítulo 3](#), é descrita a metodologia utilizada para desenvolvimento do sistema, as ferramentas e tecnologias utilizadas, os requisitos, os diversos diagramas UML criados e os aspectos relacionados ao banco de dados. No [Capítulo 4](#) são apresentados os aplicativos desenvolvidos e o resultado dos testes realizados. Por fim, no [Capítulo 5](#), são feitas as considerações finais e destacados alguns pontos que podem ser objeto de trabalhos futuros.

2 Conceitos Gerais e Revisão da Literatura

Os Sistemas Inteligentes de Transporte se baseiam na aplicação da tecnologia da informação, eletrônica e telecomunicações com o objetivo de melhorar o planejamento, gestão, operação e fiscalização do transporte público. Segundo (MEIRELLES, 1999), o número de aplicações voltadas para transportes inteligentes é amplo, incluindo sistemas de informação para usuário, gerenciamento de rodovias (sinalização de velocidade limite variável, avisos sobre eventos na pista, radares inteligentes, etc.), gerenciamento de transporte coletivo (controle de frota, gerenciamento da viagem e desvios de itinerário), controle de tráfego e de semáforo (variação do tempo em cada estado de acordo com a situação do trânsito), gerenciamento de serviços de emergência, arrecadação automática de tarifas no transporte coletivo, nos estacionamentos e nos pedágios (através de cartões e dispositivos de leitura RFID), coleta automática de dados, dentre outros.

Analizando as inúmeras aplicações existentes de SIT, verifica-se no Brasil um maior aproveitamento no controle e gerenciamento de frotas e na automatização da cobrança das tarifas. Uma área que ainda é pouco explorada é a de sistemas de informação ao usuário, por meio da qual se desenvolve uma ferramenta de diálogo entre o operador/gestor e os usuários (SCHEIN, 2003). Através desses sistemas, os usuários podem obter informações que atendam às suas demandas específicas, como tempo de espera no ponto, localização do ônibus e itinerário de determinada linha. A seguir, são apresentados alguns dos sistemas disponíveis atualmente e que fornecem informações relevantes aos utilizadores do transporte público.

O “Cadê o Ônibus?” (NANO IT, 2013) trata-se do aplicativo vencedor da hackatona realizada em 2013 pela SPTrans em parceria com a FGV. O aplicativo possibilita visualização da posição geográfica dos ônibus em tempo real, localização dos pontos de ônibus, horário de saída do terminal, itinerário dos ônibus e função de favoritos para adicionar as linhas mais utilizadas. Trata-se de um aplicativo muito completo, inclusive permite que os usuários compartilhem informações sobre o ônibus, como existência de *wifi*, ar condicionado e acessibilidade. Atualmente, atende as cidades de São Paulo, Teresina, Rio de Janeiro e Curitiba.

Desenvolvido por (BISCARO, 2014), o “BusFinder” fornece informações somente da cidade de São Paulo, e funciona com base nas informações disponibilizadas pelo serviço Olho Vivo da SPTrans. O aplicativo também fornece a posição dos ônibus em tempo real, informações das linhas com horários previstos de partida e itinerário, atualização automática e permite visualizar um número ilimitado de linhas simultaneamente.

O “DoubleMap” (DOUBLEMAP, 2015) é um aplicativo que atende as cidades

americanas de Indianápolis, Lafayette e Columbia. Ele oferece o acompanhamento da localização dos ônibus em tempo real e a rota com o itinerário a ser percorrido. Porém, a aplicação não fornece informações quanto aos horários de partida e a quantidade de linhas disponíveis é bastante limitada.

O “Take a Bus JM” ([SCHOENENKORB, 2016](#)) é um aplicativo que atende a cidade de João Monlevade. O aplicativo possui interface pouco agradável e se limita a exibição dos horários de partida dos ônibus, não exibindo a rota e a localização em tempo real.

O “Line Bus” ([EQUIPE LINE BUS, 2017](#)) é um aplicativo que fornece os horários de partida dos ônibus de algumas cidades da Bahia e de Minas Gerais, dentre elas, João Monlevade. Atualmente, está em desenvolvimento o módulo para acompanhamento da localização dos ônibus em tempo real, dessa forma, só é possível consultar o horário de partida dos ônibus. O aplicativo não disponibiliza informações da rota percorrida.

A [Tabela 1](#) destaca as características existentes em cada aplicativo mencionado e posiciona a aplicação proposta nesse trabalho com relação às dos demais aplicativos do gênero. A análise da referida tabela permite concluir que nenhum dos sistemas relacionados se adequam perfeitamente à cidade de João Monlevade/MG. Dessa forma, os aplicativos propostos nesse projeto visam oferecer ao público da cidade um sistema que contemple suas peculiaridades.

Tabela 1 – Comparativo com os sistemas relacionados

	Cadê o Ônibus?	Bus- Finder	Double- Map	Take a Bus JM	Line Bus	Sistema pro- posto
Fornecimento da localização dos ônibus em tempo real	X	X	X			X
Exibição do quadro de horários de partida	X	X		X	X	X
Exibição do quadro de horários de partida mesmo sem conexão com a internet				X	X	X
Exibição da rota percorrida pelo ônibus	X	X	X			X
Acompanhar múltiplas linhas de forma simultânea		X				X
Atende à cidade de João Monlevade				X	X	X

Fonte: Elaborado pelo autor

3 Metodologia

Neste capítulo serão apresentados os métodos e procedimentos adotados para o desenvolvimento do trabalho, a modelagem do sistema, as ferramentas e tecnologias utilizadas, os requisitos, o diagrama de caso de uso, os diagramas de atividades, os diagramas de classes e as decisões de projeto mais relevantes. Para alcançar os objetivos específicos e obter resultados de acordo com o propósito do trabalho, a metodologia segue a seguinte estrutura:

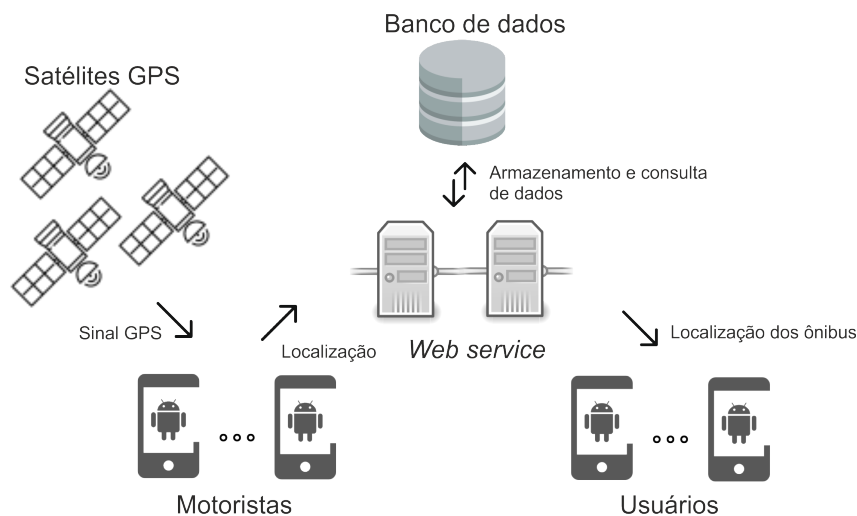
- Desenvolver um aplicativo com interface simples e intuitiva que será utilizado pelo motorista (ou cobrador) e permita selecionar a linha e o destino do ônibus e possibilitará a coleta e envio da localização para o banco de dados;
- Desenvolver um aplicativo para os usuários do transporte público que permita selecionar e visualizar os ônibus em tempo real, bem como a rota percorrida por eles. Funcionalidades adicionais como horário de partida dos ônibus e tarifas também estarão disponíveis;
- Projetar um banco de dados para armazenar as informações coletadas pelo aplicativo do motorista e disponibilizar o acesso ao aplicativo dos usuários do transporte público;
- Realizar testes para verificar o desempenho da solução proposta e identificar possíveis falhas e pontos de melhorias.

3.1 Visão geral do sistema

Um dos aplicativos, aqui referenciado como aplicativo do motorista, será responsável por coletar, de forma contínua, a coordenada geográfica dos locais por onde o ônibus estiver trafegando. Enquanto os dados são coletados, o aplicativo também deve enviá-los a um servidor, para que possam ser armazenados em um banco de dados e disponibilizados para acesso aos usuários do transporte público.

O outro aplicativo, denominado aqui como aplicativo dos passageiros, permitirá aos potenciais passageiros selecionar quais ônibus se deseja acompanhar, e deverá realizar a consulta da localização no servidor e exibir o resultado no mapa. A Figura 1 ilustra a arquitetura do sistema.

Figura 1 – Visão geral da arquitetura do sistema



Fonte: elaborado pelo autor

Para armazenamento dos dados foi criado um banco de dados relacional MySQL. A interface entre o banco de dados e os dispositivos foi feita utilizando um *web service* Java hospedado no Apache Tomcat. As requisições de envio e solicitações de dados ao servidor foram implementadas utilizando o protocolo SOAP. Essas ferramentas e tecnologias utilizadas estarão descritas na [seção 3.2](#)

3.2 Ferramentas e tecnologias utilizadas

Para o desenvolvimento dos aplicativos desse trabalho, utilizou-se o Android Studio na versão 2.3.1 para o sistema operacional Windows. O Android Studio é uma IDE que oferece suporte para a linguagem de programação Java e fornece uma série de recursos e funcionalidades que facilitam o desenvolvimento do código e a realização de testes. Além disso, permite que o aplicativo seja construído para diferentes versões do Android. Os aplicativos desenvolvidos nesse projeto estão disponíveis para os sistemas com versão 15 (Android 4.0.3) até a versão 26 (Android 8.0).

O Java trata-se de uma linguagem de programação orientada a objetos desenvolvida na década de 90 e que agora é utilizado para desenvolver aplicativos corporativos de grande porte, aprimorar a funcionalidade de servidores da *web*, fornecer aplicativos voltados ao consumo popular (por exemplo, telefones celulares, *smartphones*, *smart tvs*, etc.) e para muitos outros propósitos (DEITEL; DEITEL, 2016). Além dos aplicativos, o *web service* também foi desenvolvido utilizando a linguagem Java, e é responsável por possibilitar a integração do sistema, fazendo com que as coordenadas geográficas coletadas pelo aplicativo do motorista estejam disponíveis para consulta ao **aplicativo dos passageiros**.

Para a implementação do *web service*, utilizou-se o ambiente de desenvolvimento Eclipse Kepler, que também oferece suporte a linguagem de programação Java e permite exportar o conjunto de funcionalidades desenvolvidas como um arquivo WAR para em seguida ser feito o *deploy* no servidor. O servidor Apache Tomcat foi utilizado para possibilitar que o *web service* fique ativo, aguardando e executando as requisições.

Na comunicação entre os dispositivos móveis e o *web service* utilizou-se o protocolo SOAP, que se baseia na linguagem de marcação XML e permite a troca de mensagens estruturadas. No protocolo SOAP, uma mensagem é composta por um cabeçalho, que contém informações relevantes sobre a forma como a mensagem deve ser processada, e um corpo, que contém a mensagem a ser entregue (SNELL; TIDWELL; KULCHENKO, 2001).

Para o gerenciamento do armazenamento dos dados foi utilizado o SGBD MySQL, que utiliza a linguagem SQL como interface. O MySQL trata-se de um SGBD relacional, em que os dados são modelados e armazenados através de tabelas. Para o armazenamento das rotas dos ônibus foi utilizada a linguagem KML, que se baseia na linguagem de marcação XML e possibilita a organização de dados geográficos através de um conjunto de coordenadas compostas por latitudes e longitudes. O software Google Earth foi utilizado para transformação do itinerário textual das linhas dos ônibus em uma rota visual representativa que será exibida nos aplicativos desenvolvidos.

3.3 Especificações

Nessa seção serão apresentados os requisitos funcionais e não funcionais do projeto, o diagrama de casos de uso com o detalhamento das respectivas histórias textuais, os diagramas de atividades, os diagramas de classes e a estrutura do banco de dados utilizado para armazenamento das informações.

3.3.1 Requisitos funcionais

Os requisitos funcionais do aplicativo do motorista são:

1. Permitir que o motorista selecione a linha e o destino para poder iniciar um percurso e enviar as coordenadas geográficas;
2. Exibir um mapa com a rota prevista para a linha selecionada e mostrar a posição atual do ônibus;
3. Enviar a localização coletada para o *web service* mesmo que o aplicativo esteja em segundo plano;

4. Permitir finalizar o percurso para terminar a coleta e envio das coordenadas geográficas;
5. Possibilitar configurar a frequência de envio das coordenadas geográficas dos ônibus.

Os requisitos funcionais do **aplicativo dos passageiros** são:

1. Permitir que o usuário selecione as linhas que deseja acompanhar, exibindo no mapa a localização dos ônibus e suas rotas;
2. Possibilitar que o usuário tenha acesso aos horários de partida dos ônibus;
3. Exibir informações complementares sempre que o usuário selecionar algum dos ônibus que estão sendo atualizados em tempo real;
4. Possibilitar configurar o intervalo de atualização dos ônibus.

3.3.2 Requisitos não funcionais

Os requisitos não funcionais do sistema são:

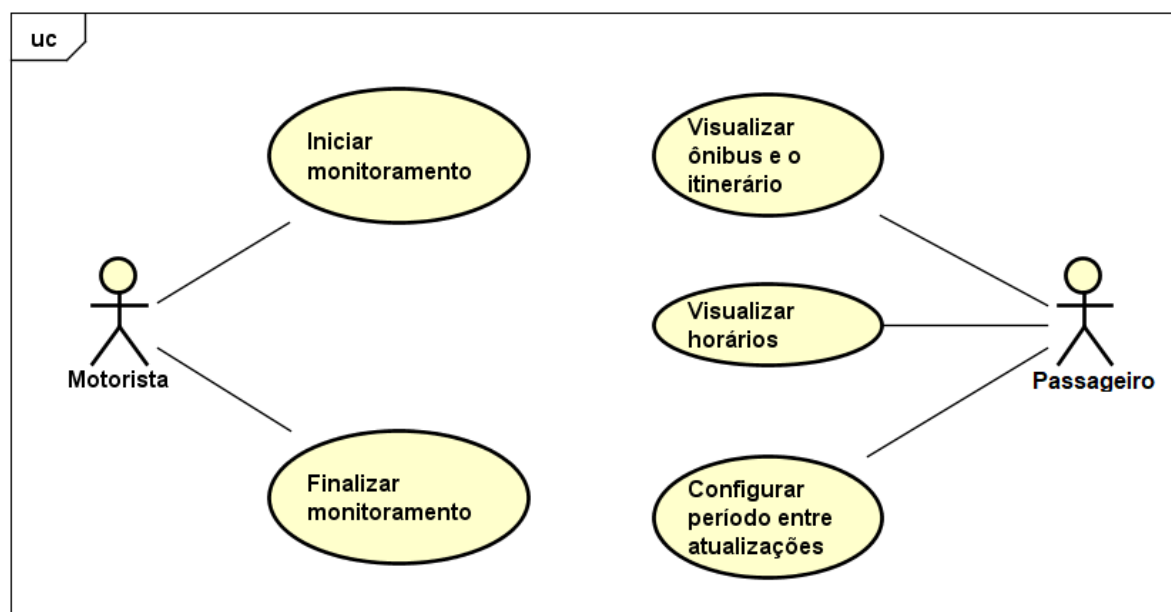
1. Funcionar perfeitamente em *smartphones* com sistema operacional Android 4.0.3 ou superior;
2. Fornecer a localização dos ônibus com confiabilidade;
3. Gerenciamento eficiente do GPS e das conexões com o banco de dados visando manter baixo o consumo de bateria e dados móveis;
4. O sistema deve funcionar perfeitamente sem muita necessidade de interação do motorista com o aplicativo.

3.4 Diagrama de Casos de Uso

O diagrama de Casos de Uso (**Figura 2**) foi desenvolvido com o apoio da ferramenta Astah. Foram definidos os seguintes atores:

- Motorista: ator responsável pelo uso do aplicativo de coleta e envio da localização dos ônibus.
- **Passageiro**: ator que representa os utilizadores do aplicativo voltado aos usuários do transporte público.

Figura 2 – Diagrama de Casos de Uso



powered by Astah

Fonte: elaborado pelo autor

À seguir, estão discriminadas as estórias textuais que representa cada um dos casos de uso. No [Quadro 1](#) está detalhado o caso de uso referente à função de iniciar o monitoramento, com os detalhes do cenário principal e do cenário alternativo.

Quadro 1 – Caso de Uso - Iniciar monitoramento

Caso de Uso: Iniciar Monitoramento
Permite ao motorista do ônibus iniciar a coleta e envio das coordenadas geográficas para o banco de dados. Ator: Motorista Pré-condições: aplicativo instalado no <i>smartphone</i> do motorista.
Cenário principal: 1. O motorista abre o aplicativo; 2. O motorista seleciona a linha e o local de partida; 3. O motorista pressiona o botão iniciar; 4. O aplicativo começa a coletar as coordenadas e enviar ao banco de dados; 5. O motorista sai da aplicação e uma notificação é criada indicando o correto envio das coordenadas geográficas. Extensões: 4a. GPS do dispositivo desabilitado 1. Caixa de diálogo é criada indicando a situação e sugerindo que seja habilitado; 2. Em caso positivo, será aberta a tela de configurações do <i>smartphone</i>. 4b. Dados móveis desabilitados 1. Caixa de diálogo é criada indicando a situação e sugerindo que seja habilitado; 2. Em caso positivo, será aberta a tela de configurações do <i>smartphone</i>. 4c. O motorista não selecionou a linha e/ou local de partida 1. O aplicativo é mantido na tela e uma mensagem informativa é exibida. 5a. Perda da conexão GPS 1. O status da notificação é atualizado indicando problema na coleta das coordenadas; 2. O aplicativo continua rodando até que a conexão seja reestabelecida e seja possível coletar e enviar as coordenadas. 5b. Perda da conexão de dados móveis 1. O status da notificação é atualizado indicando problema no envio das coordenadas; 2. O aplicativo continua rodando até que a conexão seja reestabelecida e seja possível enviar as coordenadas.

Fonte: elaborado pelo autor

Em seguida, pode ser visto no [Quadro 2](#), o detalhamento do caso de uso referente à funcionalidade de finalizar o monitoramento, que tem como ator o motorista.

Quadro 2 – Caso de Uso - Finalizar monitoramento

Caso de Uso: Finalizar monitoramento
Permite ao motorista finalizar a coleta e envio das coordenadas geográficas. Ator: Motorista Pré-condições: aplicativo instalado no <i>smartphone</i> do motorista.
Cenário principal: 1. O motorista abre o aplicativo; 2. O motorista seleciona o botão de parar o monitoramento; 3. O motorista confirma que deseja encerrar a coleta e envio das coordenadas geográficas; 4. A coleta e o envio são finalizados e a notificação é encerrada. Extensões: 3a. O motorista não confirma que deseja encerrar 1. O serviço de monitoramento não é interrompido.

Fonte: elaborado pelo autor

No [Quadro 3](#) está detalhada a estória textual referente a funcionalidade de visualização dos ônibus e rotas.

Quadro 3 – Caso de Uso - Visualizar ônibus e itinerário

Caso de Uso: Visualizar ônibus e itinerário
Permite aos usuários visualizar os ônibus que estão circulando em tempo real e suas rotas. Ator: Passageiro Pré-condições: aplicativo instalado no <i>smartphone</i>.
Cenário Principal: 1. O usuário abre o aplicativo; 2. O usuário clica no botão que exibe a lista com os ônibus; 3. O usuário seleciona os ônibus que deseja acompanhar e confirma; 4. Os ônibus em percurso são consultados no banco de dados e são exibidos no mapa juntamente com a rota que será percorrida. Extensões: 4a. Falta de conexão com a internet 1. Uma mensagem informando a impossibilidade de conectar ao servidor é exibida ao usuário. 4b. Os ônibus selecionados não estão circulando no momento 1. Uma mensagem informativa é exibida ao usuário.

Fonte: elaborado pelo autor

O [Quadro 4](#) apresenta o cenário principal da funcionalidade de visualizar os horários de partida, que tem como ator um usuário.

Quadro 4 – Caso de Uso - Visualizar horários

Caso de Uso: Visualizar horários
Permite aos usuários visualizar os horários de partida dos ônibus.
Ator: Passageiro
Pré-condições: aplicativo instalado no <i>smartphone</i> .
Cenário Principal: 1. O usuário abre o aplicativo; 2. O usuário clica no botão que exibe a lista com os ônibus; 3. O usuário clica na linha que ele deseja visualizar os horários de partida; 4. Os horários são exibidos.

Fonte: elaborado pelo autor

Por fim, o [Quadro 5](#) exibe os detalhes do caso de uso referente a funcionalidade de configuração do intervalo entre atualizações.

Quadro 5 – Caso de Uso - Configurar período entre atualizações

Caso de Uso: Configurar período entre atualizações
Permite aos usuários configurar o tempo entre as consultas no banco de dados para atualização dos dados exibidos.
Ator: Passageiro
Pré-condições: aplicativo instalado no <i>smartphone</i> .
Cenário principal: 1. O usuário abre o aplicativo; 2. O usuário clica no botão de configuração; 3. O usuário seleciona o tempo de atualização desejado.

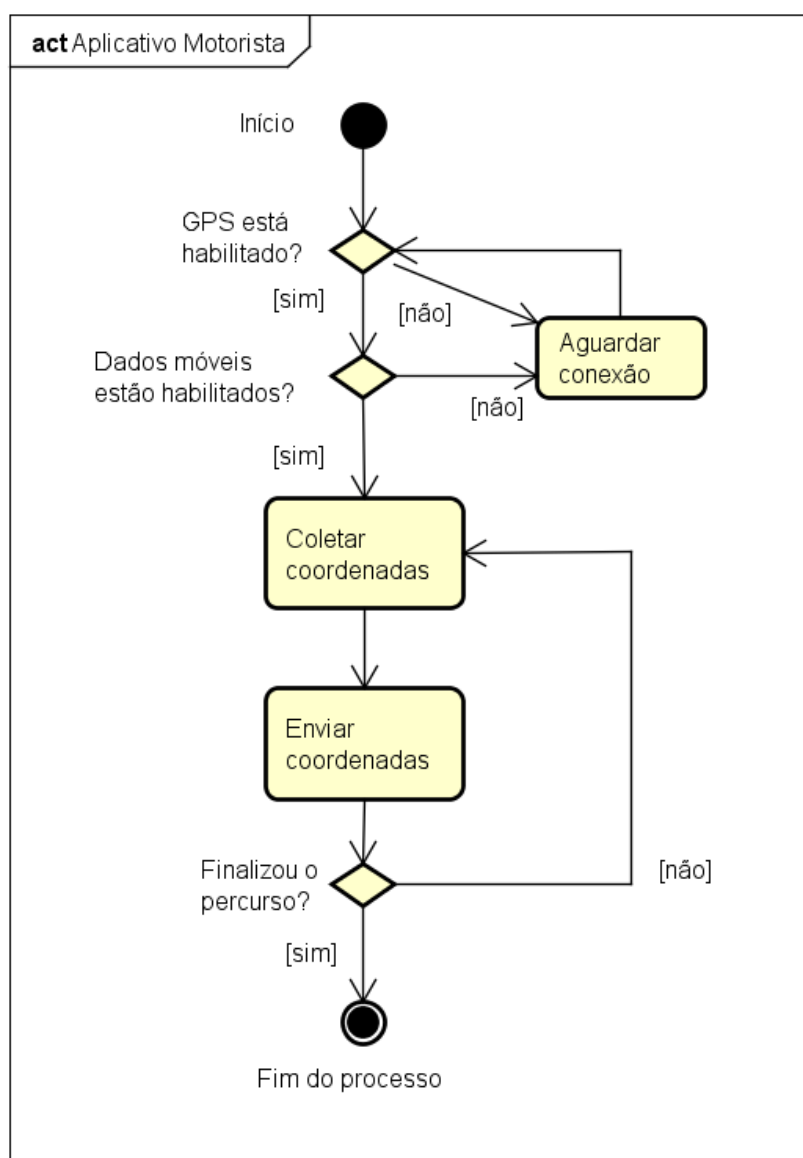
Fonte: elaborado pelo autor

Na seção seguinte serão apresentados os Diagramas de Atividades para ambos os aplicativos.

3.5 Diagramas de Atividades

A [Figura 3](#) ilustra o diagrama de atividades que representa o processo de coleta e envio das coordenadas geográficas ao banco de dados pelo aplicativo do motorista.

Figura 3 – Diagrama de Atividades do aplicativo do motorista

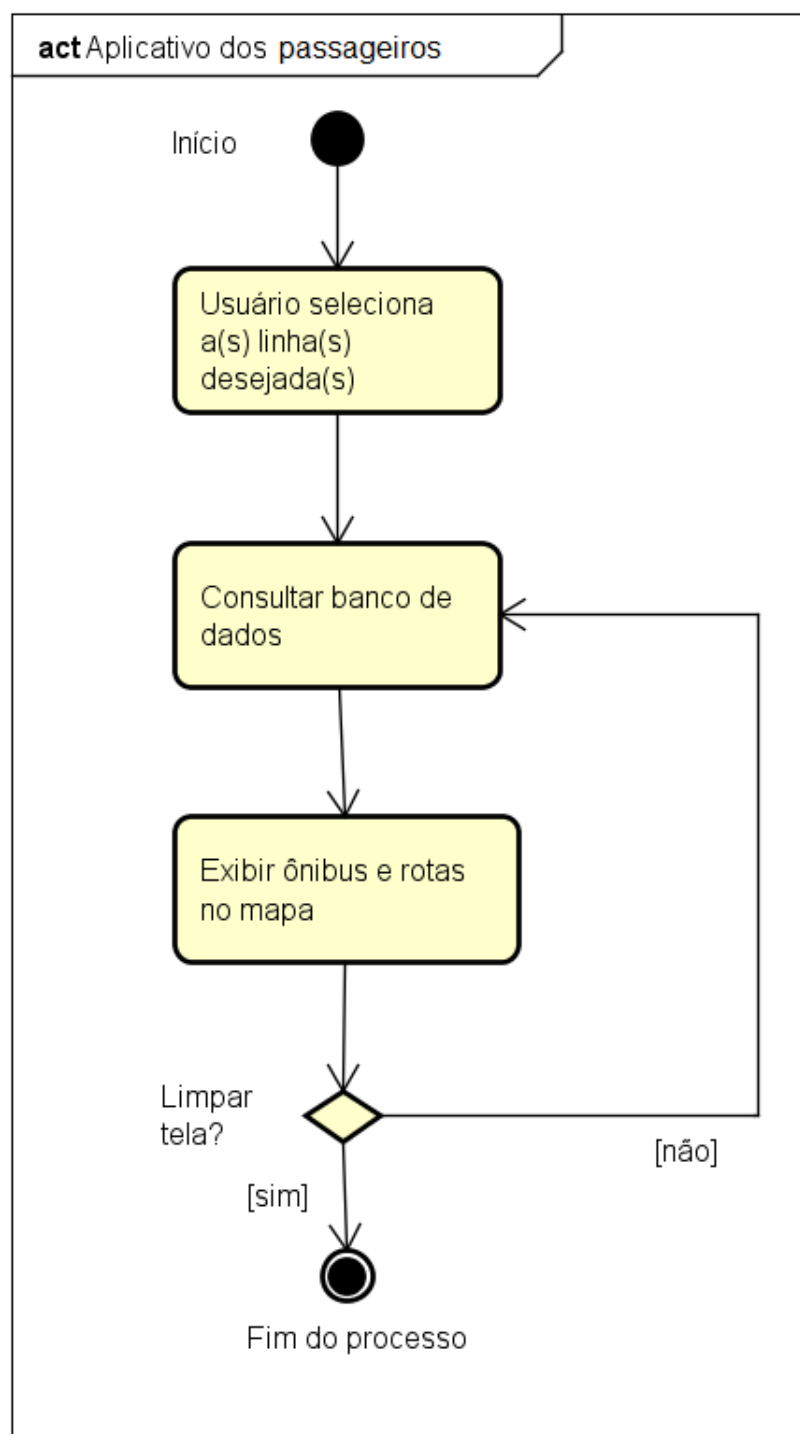


powered by Astah

Fonte: elaborado pelo autor

A [Figura 4](#) ilustra o diagrama de atividades do processo de seleção e consulta das informações de um determinado ônibus para exibição no mapa.

Figura 4 – Diagrama de Atividades do aplicativo dos passageiros



powered by Astah

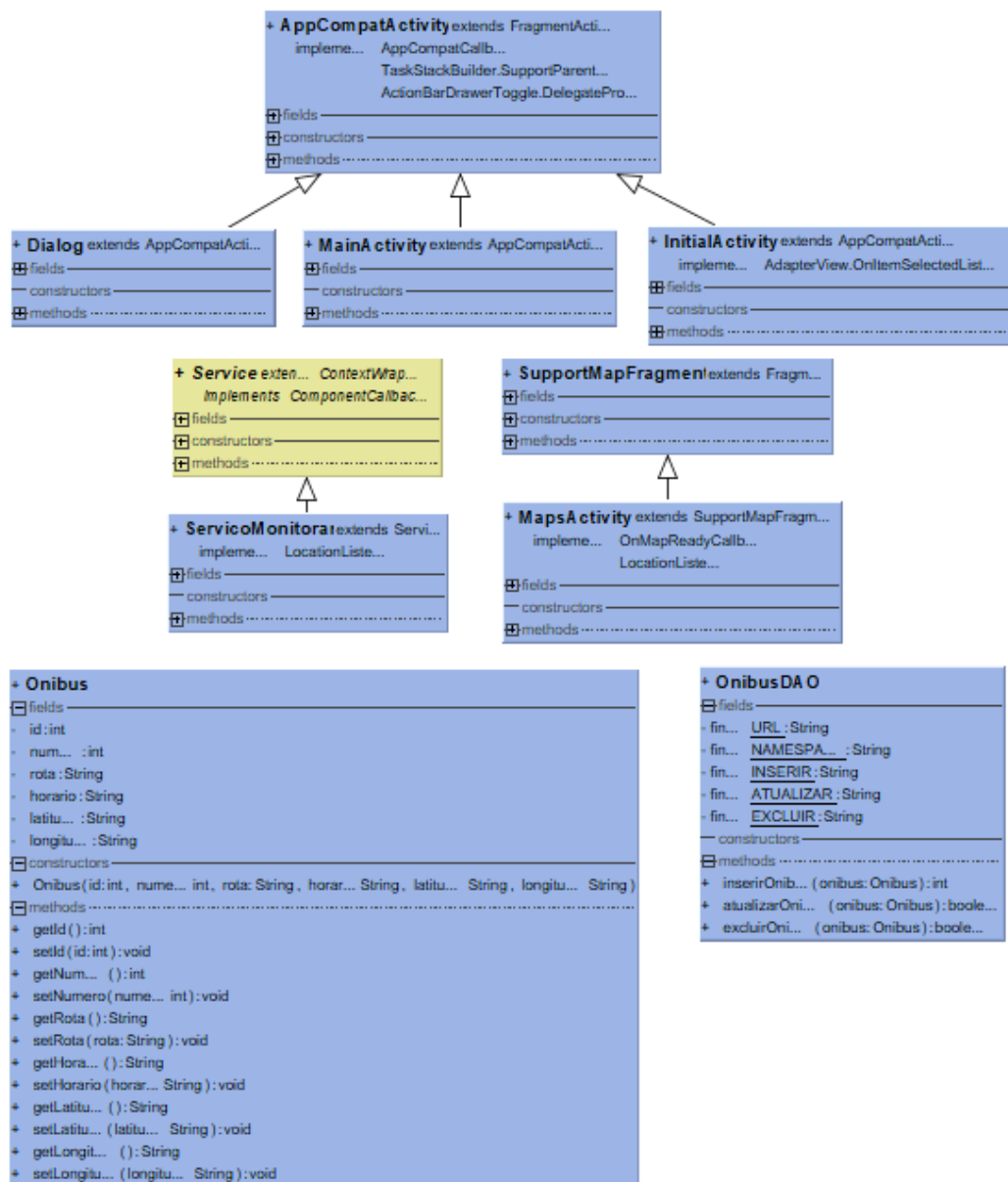
Fonte: elaborado pelo autor

A seção seguinte apresenta o Diagrama de Classes para os dois aplicativos desenvolvidos e as considerações mais importantes a respeito de cada um deles.

3.6 Diagramas de Classes

Os diagramas de classes foram gerados através do *plugin* simpleUML instalado no Android Studio. As classes para o aplicativo do motorista podem ser vistas na [Figura 5](#).

Figura 5 – Diagrama de Classes do aplicativo do motorista



Fonte: elaborado pelo autor

A seguir é descrito o propósito funcional básico de cada uma das classes.

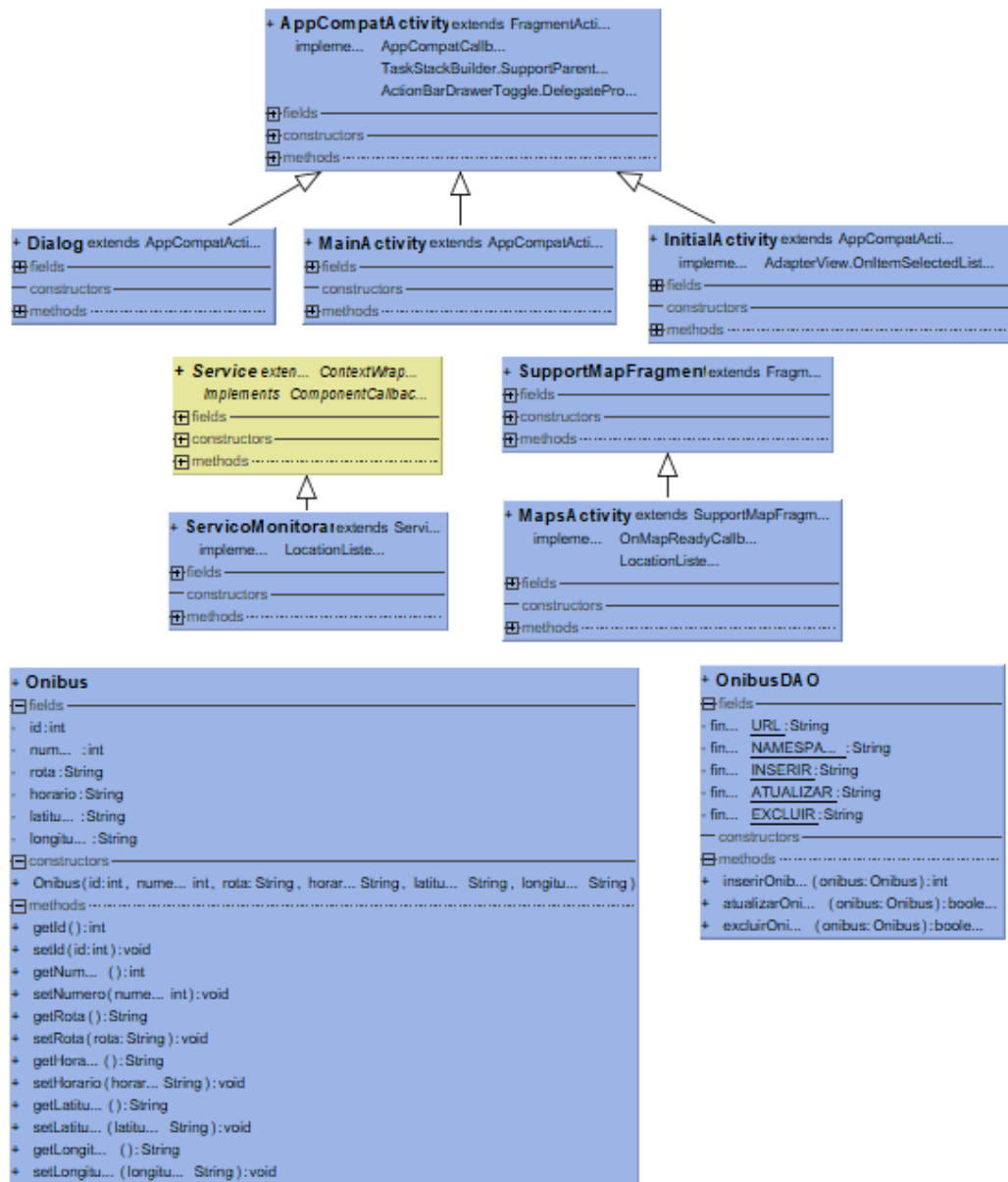
- *Onibus*: é utilizada para armazenar as informações relevantes de cada ônibus, como o identificador, o número, a rota, o horário e a posição geográfica (latitude, longitude).

- *InitialActivity*: é a classe que permite ao motorista selecionar o número e a rota do ônibus para que seja iniciado o processo de coleta e envio das coordenadas geográficas.
- *MapsActivity*: é a classe responsável pela exibição do mapa com a rota do ônibus e a posição atual.
- *ServicoMonitorar*: é a classe que permite que as coordenadas geográficas continuem sendo coletadas ainda que a aplicação esteja em segundo plano ou com a tela apagada.
- *OnibusDAO*: é a classe que fornece os métodos que permitem a comunicação com o *web service* e o armazenamento das informações no Banco de Dados.
- *MainActivity*: permite o gerenciamento dos fragmentos e a criação da notificação que indica se as coordenadas geográficas estão sendo coletadas corretamente.
- *Dialog*: possibilita a criação de caixas de diálogo personalizadas, utilizada, por exemplo, na confirmação de término do percurso.

As classes *AppCompatActivity*, *Service* e *SupportMapFragment* são nativas do Android, e são extendidas por algumas das classes desenvolvidas para possibilitar que eles herdem as particularidades necessárias.

A Figura 6 ilustra as principais classes que compõem o aplicativo dos passageiros.

Figura 6 – Diagrama de Classes do aplicativo dos passageiros



Fonte: elaborado pelo autor

Dentre as classes mais relevantes, pode-se destacar:

- *Onibus*: é utilizada para armazenar as informações relevantes de cada ônibus, como o identificador, o número, a rota, o horário e a posição geográfica (latitude, longitude).
- *OnibusDAO*: é a classe que fornece os métodos que permitem a comunicação com o *web service* e o armazenamento das informações no banco de dados.
- *MainActivity*: permite o gerenciamento dos fragmentos e a seleção das linhas que o usuário deseja acompanhar, além disso, fornece acesso ao quadro de horários.

- *MapsActivity*: é a classe responsável pela exibição do mapa com a rota do ônibus e sua localização em tempo real.
- *ConsultaServidor*: é a classe responsável por realizar periodicamente as consultas ao *web service* para obter a localização mais recente dos ônibus selecionados para acompanhamento.
- *Horario*: é a classe que permite armazenar o quadro de horários semanal para cada linha de ônibus, juntamente com os complementos, como, por exemplo, a informação de atendimento a determinado bairro em um horário específico ou a ocorrência de ponto de partida diferente do local convencional.
- *DialogAdapterConfig*: possibilita configurar o intervalo de atualização dos ônibus selecionados para acompanhamento conforme desejado (5, 15, 30, 45 ou 60 segundos).
- *DialogAdapter*: é a classe responsável pela exibição da caixa de diálogo que permite ao usuário selecionar as linhas que ele deseja acompanhar a localização em tempo real.
- *GridViewHorarios*: permite a visualização em grade dos horários dos ônibus e a exibição das informações complementares quando algum horário é selecionado.

3.7 Banco de Dados

Essencialmente, são poucas as informações que necessitam ser armazenadas no banco de dados, dessa forma, sua estrutura é bastante simplificada, tratando-se apenas de uma tabela contendo os seguintes atributos:

- Identificador: int, chave primária, auto incrementável;
- Número: int, não nulo, refere-se ao número da linha do ônibus;
- Destino: varchar(32), não nulo, refere-se ao local de destino do ônibus;
- Latitude: float, não nulo.
- Longitude: float, não nulo.

É importante destacar que tal simplicidade é imprescindível, pois se trata de uma aplicação que necessitará enviar dados frequentemente pela rede, assim, é possível que a aplicação funcione sem consumir quantidades expressivas de tráfego de dados.

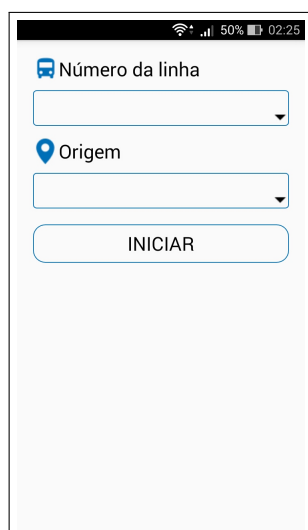
4 Apresentação e Análise dos Resultados

Neste capítulo serão apresentados os resultados obtidos com o desenvolvimento dos aplicativos, bem como o detalhamento de suas funcionalidades.

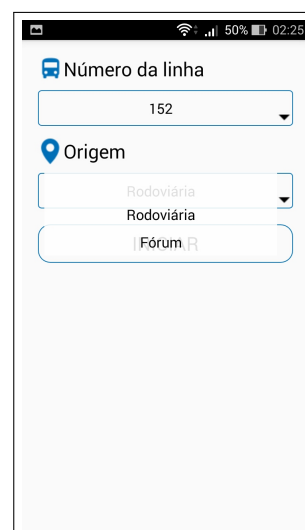
4.1 Aplicativo dos motoristas

O aplicativo desenvolvido para ser utilizado pelos motoristas dos ônibus deve oferecer uma interface bem simples, ágil e intuitiva, para que não sejam causados atrasos na partida dos ônibus pela necessidade de colocar o sistema em funcionamento. Além disso, o aplicativo deve funcionar de forma que seja diminuída ao máximo a necessidade de interação do motorista, considerando que a legislação de trânsito atual não permite essa conduta.

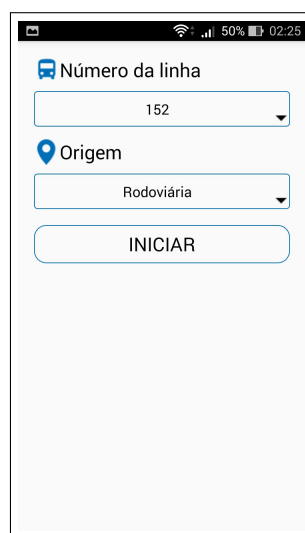
A tela inicial [Figura 7](#) permite que o motorista selecione o número da linha e o local de onde o ônibus está partindo. Se as conexões de internet e GPS estiverem habilitadas, o motorista poderá através do botão “iniciar”, dar início ao processo de envio das coordenadas geográficas ao *web service*.



(a) Antes da seleção



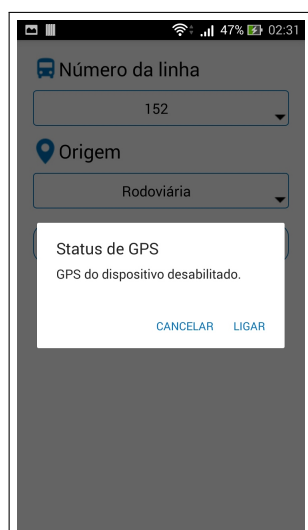
(b) Durante a seleção



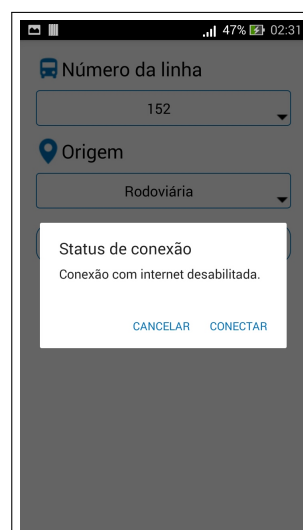
(c) Após a seleção

Figura 7 – Tela inicial do aplicativo do motorista

Caso a conexão GPS ou de internet não estejam habilitadas, uma caixa de diálogo informativa é exibida, conforme a [Figura 8](#).



(a) Conexão GPS desabilitada



(b) Conexão com a internet desabilitada

Figura 8 – Conexões desabilitadas

Caso o usuário não selecione o número da linha ou a origem, também não é possível prosseguir, conforme ilustrado na [Figura 9](#).

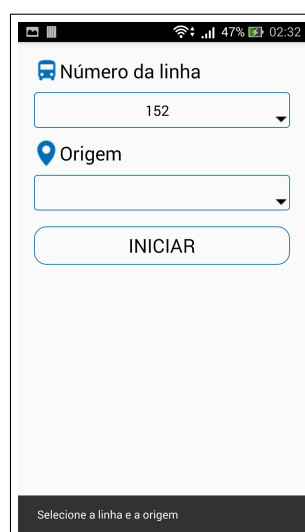


Figura 9 – Alerta para necessidade de seleção do número e da linha do ônibus

Quando o processo de coleta e envio das coordenadas geográficas é iniciado, é exibido no mapa a rota da linha previamente selecionada e a posição atual do ônibus. O aplicativo sempre mantém a tela centralizada na posição atual do ônibus, assim, se houver necessidade, o motorista poderá fixar o *smartphone* em um local seguro e acompanhar a rota prevista da linha selecionada. Se o aplicativo for colocado em segundo plano ou a tela for apagada, as coordenadas geográficas continuam sendo coletadas e enviadas.

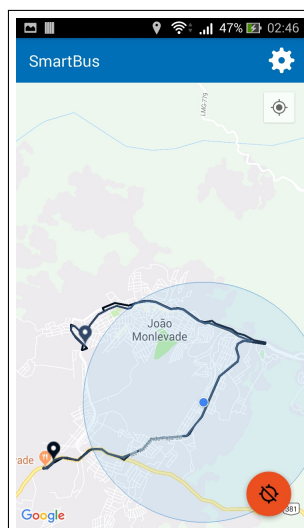


Figura 10 – Mapa com a rota e posição atual do ônibus

Uma notificação indicando o *status* de envio das coordenadas é criada e mantida na barra de notificações (Figura 11). Através do botão “finalizar” existente na notificação ou do botão flutuante existente no canto inferior direito do mapa (Figura 10) é possível encerrar a coleta e envio das coordenadas geográficas. Em ambos casos, é solicitada a confirmação de término do percurso, conforme a Figura 12.

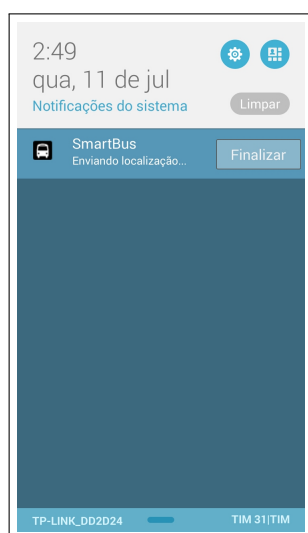
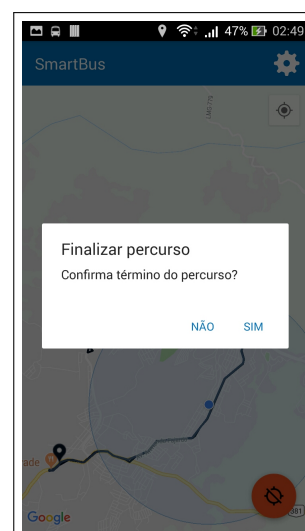
Figura 11 – Notificação com *status*

Figura 12 – Confirmação de término do percurso

Por meio do botão de engrenagem, localizado na toolbar do aplicativo, é exibida uma caixa de diálogo, conforme ilustrado na Figura 13, onde é possível definir o intervalo de coleta e envio das coordenadas. O tempo padrão entre envios é de 15 segundos, sendo possível alterar para 5, 30, 45 ou 60 segundos.

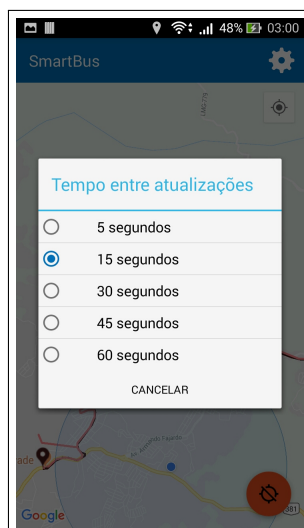


Figura 13 – Caixa de diálogo para definição do intervalo entre envios

Na seção seguinte é apresentado o aplicativo dos usuários e suas funcionalidades.

4.2 Aplicativo dos passageiros

O aplicativo direcionado aos usuários do transporte público deve oferecer uma interface atraente e intuitiva, considerando que pessoas de diversas faixas etárias poderão utilizá-lo. As cores foram escolhidas com o objetivo de prover um design agradável e também fazer alusão à cor predominante dos ônibus que circulam na cidade de João Monlevade.

A tela inicial do aplicativo voltado aos usuários do transporte público é uma *splash screen* indicando o nome do sistema e a cidade, conforme a Figura 14. Em seguida, é exibida a tela com o mapa centrado na posição atual do usuário, indicada pelo círculo azul, como pode ser visto na Figura 15. Caso o GPS do dispositivo esteja desabilitado, o mapa é centrado na cidade de João Monlevade. É possível alterar a área visualizada de acordo com a necessidade, bastando tocar na tela e mover para ajustar a posição desejada.

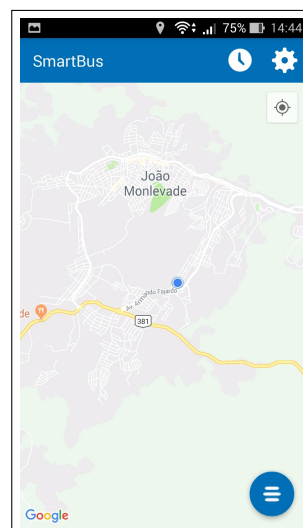
Figura 14 – *Splash Screen*

Figura 15 – Mapa indicando a posição do usuário

Utilizando o botão que aparece no canto superior direito do mapa é possível atualizar a posição do usuário, fazendo com o que o mapa fique centralizado nesse local. Através do botão flutuante, localizado na parte inferior direita da tela, é exibida a lista de linhas para que o usuário possa selecionar as que deseja acompanhar em tempo real, conforme a [Figura 16](#).

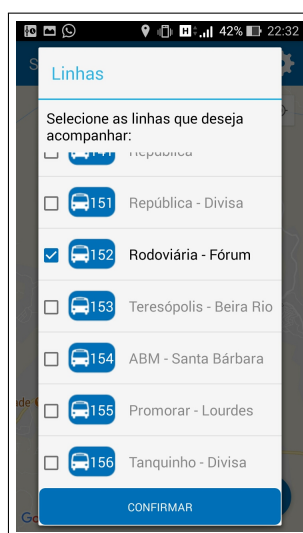
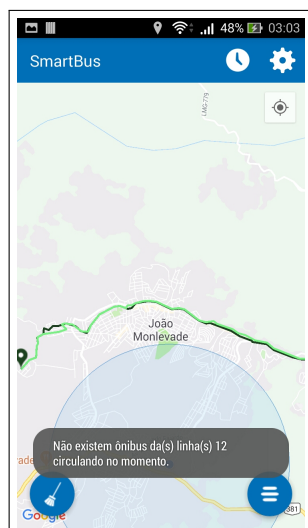
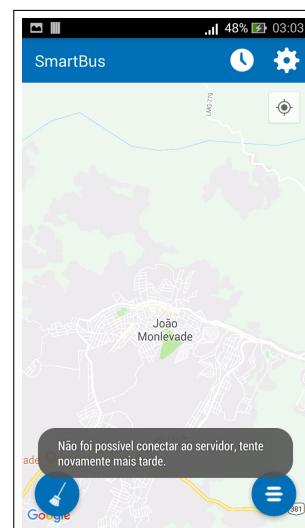


Figura 16 – Tela com lista de linhas para acompanhamento

Em caso de não existirem ônibus da linha selecionada circulando no momento, problemas com o servidor ou falta de conexão com a internet, mensagens informativas das situações são exibidas ao usuário, como pode ser visto na [Figura 17](#).



(a) Ausência de ônibus circulando



(b) Impossibilidade de conexão com o servidor

Figura 17 – Mensagens informativas

Após selecionar as linhas que deseja acompanhar, as rotas dessas linhas são exibidas no mapa, juntamente com os marcadores de ponto de partida e destino, e são feitas consultas ao banco de dados para se obter a localização dos ônibus. Em seguida, os marcadores que representam os ônibus são adicionados ao mapa e as atualizações feitas periodicamente, como ilustrado na [Figura 18](#).

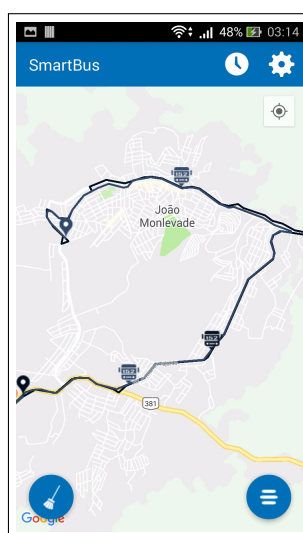
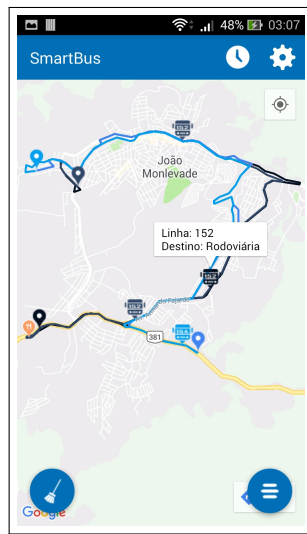


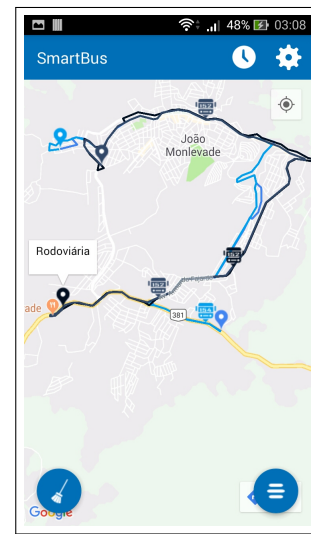
Figura 18 – Rota e localizações exibidas após seleção

Foram definidas duas cores distintas para representar os ônibus em ambos os sentidos, isto é, ida e volta. Ao clicar no ícone de algum dos ônibus que esteja sendo mostrado, é possível verificar em detalhes o número da linha e o destino. Além disso, caso o usuário selecione mais de uma linha para acompanhar simultaneamente, as rotas e

localizações são mostradas em cores distintas, para facilitar a visualização. Esses detalhes podem ser vistos na [Figura 19](#).



(a) Detalhes do ônibus selecionado



(b) Detalhe do ponto de destino selecionado

Figura 19 – Duas linhas sendo exibidas simultaneamente

Quando existem linhas selecionadas para acompanhamento, um novo botão flutuante aparece no canto inferior esquerdo da tela, e pode ser utilizado pelo usuário para limpar o mapa, removendo todas as linhas selecionadas previamente. Nesse caso, é exibida uma caixa de diálogo solicitando confirmação do usuário, conforme a [Figura 20](#).

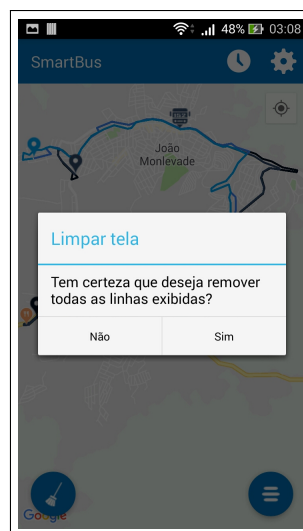


Figura 20 – Confirmação para limpar o mapa

Por meio do botão de engrenagem, localizado na *toolbar* do aplicativo, é exibida uma caixa de diálogo ([Figura 21](#)) onde é possível selecionar o intervalo de atualização desejado. O tempo padrão para atualização é de 15 segundos, sendo possível alterar para 5, 30, 45 ou 60 segundos, de acordo com a preferência do usuário.

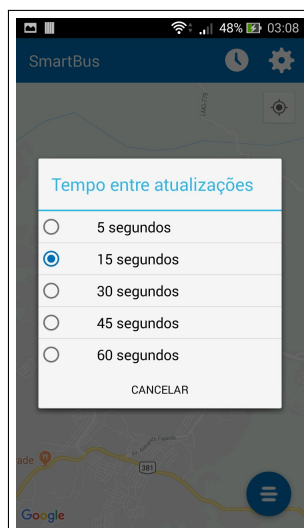


Figura 21 – Caixa de diálogo para seleção do intervalo de atualização desejado

Utilizando o botão cujo símbolo é um relógio, também localizado na toolbar do aplicativo, é exibida a lista de linhas de ônibus disponíveis para que o usuário possa escolher uma para visualizar os horários, conforme ilustrado na [Figura 22](#).

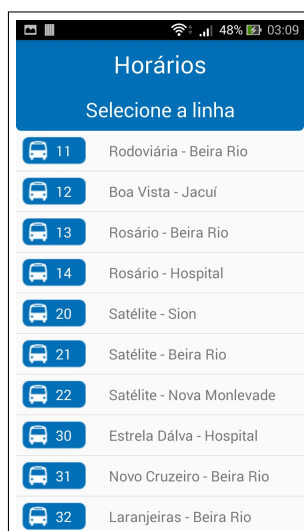


Figura 22 – Lista de linhas de ônibus

Em seguida, é exibida a tela contendo os horários de partida dos ônibus da linha selecionada. Os horários foram organizados em três guias: “dias úteis”, “sábados” e “domingos e feriados”. A guia exibida por padrão é definida de acordo com o momento em que a aplicação está executando, ou seja, caso seja sexta-feira a guia “dias úteis” é automaticamente selecionada, caso seja domingo, a guia “domingos e feriados” é selecionada por padrão. O usuário pode alternar entre as guias conforme desejar. Para cada guia é exibido o quadro de horários referente ao ponto de partida do ônibus, conforme a [Figura 23](#).

Horários				
14				
Rosário - Hospital				
Clique no horário para exibir informações complementares				
DIAS ÚTEIS	SÁBADOS	DOMINGOS E FERIADOS		
ROSÁRIO				
08:10	10:00	11:25	12:40	20:00
HOSPITAL				
05:50	10:50	12:00	13:25	15:30
18:10	20:35	22:25		

Figura 23 – Quadro de horários de partida

Para ver informações complementares referentes a algum horário de partida (bairros atendidos ou o local da partida diferente do convencional) o usuário pode pressionar sobre o horário desejado. Caso não existam informações a serem exibidas, uma mensagem informando tal situação é apresentada ao usuário. Esses recursos podem ser vistos na [Figura 24](#).

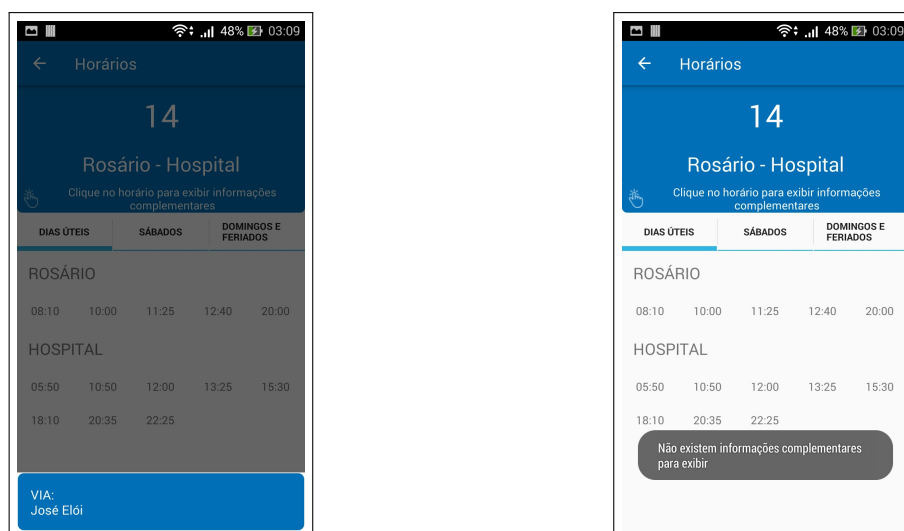


Figura 24 – Exibição das informações complementares

Na seção seguinte são apresentados alguns testes que foram realizados com os aplicativos e os resultados que foram observados.

4.3 Testes realizados

Para verificar o funcionamento do sistema proposto nesse trabalho foram utilizados dois *smartphones*, um Asus Zenfone 5 com sistema operacional Android 4.4.2 em que foi instalado o aplicativo do motorista, e um Motorola Moto G4 com sistema operacional Android 7.1.1 no qual foi instalado o **aplicativo dos passageiros**. O *web service* ficou hospedado em um desktop CCE com processador Intel Core i3-2100, 4 GB de memória RAM e sistema operacional Windows 7 SP3.

A precisão das coordenadas fornecidas pelo *smartphone* e as funcionalidades gerais do sistema foram testadas com o auxílio de uma motocicleta. Partindo-se da rodoviária da cidade de João Monlevade foi realizado o trajeto da linha 152 até o ponto final próximo ao Fórum Milton Campos. Durante o trajeto percorrido pode-se obter, com sucesso, as informações do monitoramento e a precisão obtida foi muito satisfatória. O consumo de bateria e dados móveis se mantiveram em níveis aceitáveis, demonstrando que os cuidados adotados durante a implementação resultaram nos efeitos desejados. Também foram realizados testes em bairros mais afastados do centro urbano, e em determinados locais, houve um pequeno atraso na transmissão dos dados via conexão móvel, devido a oscilações na conexão por deficiência da cobertura nesses locais. No entanto, o atraso foi pouco significativo, não comprometendo o comportamento do sistema.

Além disso, foram realizados testes para identificar possíveis divergências no *layout* das aplicações e o funcionamento das demais funcionalidades dos aplicativos em diversas versões do Android. Os testes foram realizados através da criação de vários dispositivos virtuais com diferentes tamanhos de telas e sistema operacional Android entre as versões 4.0.3 e 8.1. Em todos os casos, o *layout* se manteve conforme o esperado e os recursos funcionaram de forma adequada, além disso, os aplicativos executaram sem ocorrência de travamentos e erros.

5 Conclusões e Trabalhos Futuros

Com base no trabalho desenvolvido pode-se concluir que, por meio da combinação de tecnologias como a localização via satélite e o uso de *smartphones*, é possível desenvolver um sistema eficiente de monitoramento, com precisão satisfatória e baixo investimento. As tecnologias e ferramentas selecionadas para desenvolvimento do trabalho foram adequadas e contribuíram para o desempenho satisfatório do sistema.

A cidade de João Monlevade possui uma quantidade significativa de linhas de ônibus, e, apesar de se tratar de uma cidade de pequeno porte, conta com duas universidades públicas. Dessa forma, o sistema de transporte público acaba sendo bastante utilizado, principalmente nos horários de pico, e a cidade carece de um sistema que forneça a localização dos ônibus em tempo real.

Apesar de não ter sido possível realizar testes com complexidade e semelhança a uma carga real no sistema, devido à impossibilidade de simular vários ônibus em deslocamento simultaneamente, os resultados obtidos demonstraram que o sistema está funcional e tem potencial para implantação.

A principal limitação para implementação imediata do sistema na cidade de João Monlevade está na dependência da coleta de dados estar atrelada a execução de um aplicativo por parte do motorista do ônibus. Essa limitação pode ser superada através do uso de sistemas de rastreamento baseados na tecnologia GPRS. Algumas empresas de transporte público já fazem uso dessa tecnologia para controlar a sua frota; porém, a exemplo da empresa que atua em João Monlevade, a grande maioria não disponibiliza esses dados para os usuários do serviço, conforme proposto no presente trabalho. Ainda assim, vale destacar a pertinência da solução proposta, que pode ser aproveitada em cidades nas quais as empresas provedoras de serviços de transporte público não utilizam nenhum sistema de rastreamento, uma vez que ela tende a ser mais econômica que a implantação de outros sistemas de monitoramento que necessitam de *hardwares* e arquiteturas de comunicação específicas.

Os trabalhos futuros podem incluir:

- Aperfeiçoamento dos aplicativos desenvolvidos, com inclusão de novas funcionalidades, tais como, exibição de pontos de ônibus no mapa (com o detalhamento das linhas que atendem ao ponto), possibilidade de favoritar linhas utilizadas com frequência, criação de alertas e exibição de notícias e avisos sobre o sistema de transporte público da cidade.
- Utilização de mecanismos de inteligência artificial e mineração de dados para extrair

informações relevantes, como intensidade do trânsito por dia da semana e horário, e também predição do tempo de deslocamento.

- Inclusão de mecanismos que permitam aos usuários reportar o nível de satisfação com o transporte público que é oferecido.
- Inserção de funcionalidade que permita ao usuário adicionar as informações do destino até onde deseja se deslocar. Com base nessas informações, fornecer os ônibus que atendem ao destino.
- Verificação da viabilidade de implementar o aplicativo dos passageiros de forma colaborativa, de forma que ele possa ser utilizado para fornecer a localização dos ônibus, visando não necessitar de um aplicativo exclusivo com esse propósito.
- Desenvolvimento de versões para a plataforma iOS dos aplicativos do motorista e dos usuários.
- Realização de contato com a prefeitura municipal para verificar a possibilidade de acordo com a empresa responsável pelo transporte público na cidade de João Monlevade para desenvolvimento de uma interface que forneça os dados do sistema GPRS ao aplicativo proposto nesse trabalho.
- Extensão do sistema para atendimento a outras cidades.

Por fim, pode-se concluir que todos os objetivos propostos nesse trabalho foram alcançados e a realização de trabalhos futuros poderá contribuir para melhora contínua do sistema sugerido.

Referências

ANDROID. *Developers Android Studio*. Disponível em: <<https://developer.android.com/about/>>. Nenhuma citação no texto.

APACHE TOMCAT. *Documentation Index*. Disponível em: <<http://tomcat.apache.org/tomcat-8.0-doc/>>. Nenhuma citação no texto.

BISCARO, M. *BusFinder*. 2014. Disponível em: <https://play.google.com/store/apps/details?id=com.marcobiscaro.busfinder.activities&hl=pt_BR>. Citado na página 17.

DEITEL, H. M.; DEITEL, P. J. *Java Como Programar: 10ª edição*. [S.l.]: Pearson, 2016. Citado na página 21.

DOUBLEMAP. *ROCHESTER PUBLIC TRANSIT*. 2015. Disponível em: <<https://www.doublemap.com/rochester-case-study/>>. Citado na página 17.

EQUIPE LINE BUS. *Line Bus*. 2017. Disponível em: <<https://linebusapp.com.br/>>. Citado na página 18.

GOOGLE. *Keyhole Markup Language*. Disponível em: <<https://developers.google.com/kml/>>. Nenhuma citação no texto.

LECHETA, R. R. *Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. [S.l.]: Novatec, 2013. Nenhuma citação no texto.

MEIRELLES, A. A. de C. *Sistemas de transporte inteligentes: aplicação da telemática na gestão do trânsito urbano*. v. 1, 1999. Disponível em: <http://www.ip.pbh.gov.br/ANO1_N1_PDF/ip0101meirelles.pdf>. Citado na página 17.

MEIRELLES, F. S. 29ª pesquisa anual do uso de ti. In: FUNDAÇÃO GETÚLIO VARGAS. 2018. Disponível em: <<https://eaesp.fgv.br/ensinoeconhecimento/centros/cia/pesquisa>>. Citado na página 15.

MYSQL. *MySQL Documentation*. Disponível em: <<https://dev.mysql.com/doc/>>. Nenhuma citação no texto.

NANO IT. *Cadê o Ônibus*. 2013. Disponível em: <<http://www.cadeoonibus.com.br/CoO/SiteV2>>. Citado na página 17.

ORACLE. *Oracle Documentation*. Disponível em: <<http://www.oracle.com/technetwork/java/index.html>>. Nenhuma citação no texto.

SCHEIN, A. L. *Sistema de Informação ao usuário como estratégia de fidelização e atração*. 2003. Citado na página 17.

SCHOENENKORB, F. *Take a Bus JM*. 2016. Disponível em: <https://play.google.com/store/apps/details?id=takeabus.schoenenkorb.takeabus.monlevade&hl=pt_BR>. Citado na página 18.

SNELL, J.; TIDWELL, D.; KULCHENKO, P. *Programming web services with SOAP: building distributed applications*. [S.l.]: O'Reilly Media, Inc, 2001. 13–22 p. Citado na página [22](#).

WIGGERS, C. et al. *Professional Apache Tomcat*. [S.l.]: John Wiley & Sons, 2004. Nenhuma citação no texto.