

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

BLUMENBUS: APLICATIVO MÓVEL PARA CONSULTA DE
INFORMAÇÕES DO TRANSPORTE PÚBLICO DE
BLUMENAU

MARCELA ZANATTA KOEPEL

BLUMENAU
2018

MARCELA ZANATTA KOEPEL

**BLUMENBUS: APLICATIVO MÓVEL PARA CONSULTA DE
INFORMAÇÕES DO TRANSPORTE PÚBLICO DE
BLUMENAU**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof^a Luciana Pereira de Araújo Kohler, Mestra - Orientadora

**BLUMENAU
2018**

**APLICATIVO MÓVEL PARA CONSULTA DE
INFORMAÇÕES DO TRANSPORTE PÚBLICO DE
BLUMENAU**

Por

MARCELA ZANATTA KOEPEL

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

Presidente: Luciana P. de Araújo Kohler
Prof(a). Luciana Pereira de Araújo Kohler – Orientador, FURB

Membro: Gilvan Justino
Prof(a). Gilvan Justino – FURB

Membro: Francisco Adell Péricas
Prof(a). Francisco Adell Péricas – FURB

Blumenau, 12 de dezembro de 2018

Dedico este trabalho à minha família, que sempre apoiou minhas escolhas.

AGRADECIMENTOS

Aos meus pais Werner Koepsel e Jacqueline Maria Zanatta Koepsel pelo apoio constante.

Ao meu irmão Júlio Werner Zanatta Koepsel, por compartilhar meu gosto por tecnologia.

Ao meu namorado Diogo Vinicius Junges Krutzsch, pela paciência e compreensão nos dias mais difíceis.

Aos meus amigos pelo suporte durante os anos de faculdade.

À minha orientadora, Luciana Pereira de Araújo Kohler, pela excelente orientação.

A todos que estiveram envolvidos de alguma maneira na realização deste trabalho.

Só se pode alcançar um grande êxito quando
nos mantemos fiéis a nós mesmos.

Friedrich Nietzsche

RESUMO

Este trabalho apresenta a especificação, desenvolvimento e operacionalidade de um aplicativo móvel colaborativo que tem por objetivo fornecer informações referentes ao transporte público da cidade de Blumenau, em Santa Catarina. As informações são relacionadas às linhas de ônibus disponíveis, seus horários de partida, pontos de saída e trajeto realizado. A aplicação promove o compartilhamento de informações através de comentários realizados pelos usuários no aplicativo sobre as linhas do transporte público, objetivando a disseminação do conhecimento em tempo real. O aplicativo, denominado BlumenBus, foi desenvolvido com a utilização da biblioteca Ionic sobre a plataforma Cordova, na linguagem de programação JavaScript através do *superset* TypeScript e *framework* Angular, e com a utilização do serviço de *backend* do Firebase, utilizando seu banco de dados não relacional. O aplicativo desenvolvido foi testado por dois avaliadores especialistas através de questionário de avaliação heurística, afim de aferir o grau de usabilidade da aplicação que mostrou-se adequada dentro dos padrões de usabilidade conhecidos.

Palavras-chave: Transporte público. Colaborativo. Aplicativo móvel.

ABSTRACT

This work presents the specification, development and operationalization of a collaborative mobile application that aims to provide information about public transportation in the Blumenau city, Santa Catarina state. The information is about available bus lines, departure times, departure points and routes. The application promotes the sharing information through comments do by users about public transport lines, aiming the dissemination of knowledge in real time. The application, called BlumenBus was developed using the Ionic library on the Cordova platform, in the JavaScript programming language using the superscript TypeScript and Angular framework. The Firebase backend service was used its non-relational database and user authentication tool. The developed application was tested by two expert evaluators through a heuristic evaluation questionnaire in order to assess the degree of usability of the application.

Key-words: Public transportation. Collaborative. Mobile application.

LISTA DE FIGURAS

Figura 1 - Modelo 3C de colaboração	17
Figura 2 - Seleção para pesquisa	19
Figura 3 - Horários com observação de desvio	20
Figura 4 - Trajeto no mapa	21
Figura 5 - Diagrama de casos de uso	24
Figura 6 - Estrutura inicial.....	26
Figura 7 - Estrutura das linhas.....	27
Figura 8 - Console do Firebase.....	29
Figura 9 - Serviço de autenticação	37
Figura 10 - Diagrama de atividades da consulta ao trajeto da linha.....	39
Figura 11 - Telas principais do aplicativo	40
Figura 12 - Telas de horários e mapa	41
Figura 13 - Distribuição dos problemas encontrados	46
Figura 14 - Introdução ao questionário.....	52
Figura 15 - Objetivos da aplicação do questionário	52
Figura 16 - Termo de consentimento do questionário.....	53
Figura 17 - Parte dois do termo de consentimento	53
Figura 18 - Avaliação Heurística.....	54
Figura 19 - Heurística H1	55
Figura 20 - Heurística H2	55
Figura 21 - Heurística H3	56
Figura 22 - Heurística H4	56
Figura 23 - Heurística H5	57
Figura 24 - Heurística H6	57
Figura 25 - Heurística H7	58
Figura 26 - Heurística H8	58
Figura 27 - Heurística H9	59
Figura 28 - Heurística H10	59
Figura 29 - Heurística H11	60
Figura 30 - Heurística H12	60
Figura 31 - Heurística H13	61

LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais e a matriz de rastreabilidade dos UCs.....	23
Quadro 2 - Requisitos Não Funcionais.....	23
Quadro 3 - HTML da tela do mapa	30
Quadro 4 - HTML da tela de comentários.....	31
Quadro 5 - Script da tela de informações da linha	32
Quadro 6 - Variáveis SASS	33
Quadro 7 - Método de inicialização do mapa.....	34
Quadro 8 - Conexão com a base de dados.....	36
Quadro 9 - Métodos de autenticação	37
Quadro 10 - Script da tela de cadastro.....	38
Quadro 11 - Comparativo trabalhos correlatos.....	42
Quadro 12 - Resumo das heurísticas, problemas e gravidade	45
Quadro 13 - Problemas e justificativas.....	46

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

BAAS – Backend as a Service

CSCW – Computer Supported Cooperative Work

CSS – Cascading Style Sheets

HTML – Hypertext Markup Language

JSON – JavaScript Object Notation

RF – Requisito Funcional

RNF – Requisito Não Funcional

RS – Rio Grande do Sul

SASS - Syntactically Awesome Style Sheets

SC – Santa Catarina

SETERB – Serviço Autônomo Municipal de Trânsito e Transportes de Blumenau

SDK – Software Development Kit

UC – Use Case (Caso de Uso)

URL - Uniform Resource Locator

WAP – Wireless Application Protocol

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 TRANSPORTE COLETIVO NA CIDADE DE BLUMENAU	15
2.2 SISTEMAS COLABORATIVOS	16
2.3 FERRAMENTAS DE APOIO PARA SISTEMAS COLABORATIVOS	17
2.4 TRABALHOS CORRELATOS.....	18
2.4.1 BluBus.....	18
2.4.2 MovelBus.....	19
2.4.3 BusMaps.....	20
3 DESENVOLVIMENTO	22
3.1 LEVANTAMENTO DE INFORMAÇÕES.....	22
3.2 ESPECIFICAÇÃO	22
3.2.1 Diagrama de Casos de uso	23
3.2.2 Estrutura do banco de dados não relacional.....	26
3.3 IMPLEMENTAÇÃO	27
3.3.1 Técnicas e ferramentas utilizadas.....	27
3.3.2 Desenvolvimento da aplicação.....	29
3.3.3 Operacionalidade da implementação	38
3.4 RESULTADOS E DISCUSSÕES.....	41
3.4.1 Comparação com os trabalhos correlatos.....	41
3.4.2 Avaliação heurística	43
3.4.3 Resultados	45
4 CONCLUSÕES.....	47
4.1 EXTENSÕES	48
REFERÊNCIAS	50
APÊNDICE A – FORMULÁRIO DE AVALIAÇÃO	52

1 INTRODUÇÃO

O sistema de transporte público brasileiro sempre recebeu, e continua recebendo, um grande número de reclamações devido à baixa qualidade do serviço prestado (PENA, 2013). Devido a necessidade de disponibilização de transporte para os cidadãos e a falta de meios das Prefeituras Municipais para gerir por conta própria um sistema de transporte coletivo, a prestação deste serviço foi terceirizada para empresas privadas que, historicamente, buscam maximizar seus resultados a partir da diminuição dos gastos (PENA, 2013).

Um ponto de impacto relevante para a decadência da qualidade do transporte coletivo público vem a partir das iniciativas do governo Federal para dar um maior acesso para a população aos veículos próprios e individuais. Aliada a falta de um planejamento para a mobilidade dos municípios, tal fato acabou também por desencadear a piora na situação do trânsito nas cidades (PENA, 2013).

De acordo com estudos realizados por Ferris (2011), em pesquisas sobre a aplicação OneBusAway (aplicação que disponibiliza as informações do sistema de transporte coletivo na cidade de Seattle e região, no estado de Washington nos Estados Unidos da América) 91% responderam que ao utilizar a aplicação gastaram menos tempo esperando pelo ônibus. Em decorrência da análise sobre os dados coletados, foi notado que a satisfação média dos usuários do transporte coletivo estava fortemente ligada ao menor tempo de espera. Procura-se a utilização de inovações tecnológicas já presentes na rotina da população para melhorar a qualidade do serviço de transporte público.

Segundo pesquisa realizada pelo CETIC (2016), na região sul do Brasil, 88 % das pessoas entrevistadas possuem telefone celular. Paralelamente em outra pesquisa similar realizada pelo mesmo centro foi observado que 84% das pessoas da região sul responderam que costumam utilizar a internet através do aparelho celular todos os dias ou quase todos os dias.

Diante deste cenário, este trabalho apresenta o desenvolvimento de um aplicativo móvel que auxilie os usuários do transporte coletivo de Blumenau a realizarem a consulta de informações referentes ao serviço. O aplicativo também visa oferecer um meio de colaboração entre os usuários, a partir da troca de informações e experiências sobre as linhas de ônibus utilizadas.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um aplicativo móvel para disponibilizar informações do transporte público.

Os objetivos específicos são:

- a) fornecer uma interface para colaboração entre os usuários do aplicativo;
- b) disponibilizar interfaces gráficas mantendo requisitos e padrões de usabilidade aplicados a dispositivos móveis com base nas diretrizes do Android e do Material Design.

1.2 ESTRUTURA

Esta monografia está organizada de forma a descrever a pesquisa e o desenvolvimento do aplicativo desenvolvido, dividida em quatro capítulos.

Além do capítulo introdutório, em que é realizada uma introdução e são apresentados os objetivos deste trabalho, tem-se o capítulo 2 no qual é apresentada a fundamentação teórica, discorrendo sobre a situação do transporte coletivo na cidade de Blumenau, a relevância dos sistemas colaborativos e ferramentas que apoiem estes sistemas, bem como os trabalhos correlatos. O capítulo 3 apresenta o desenvolvimento do aplicativo BlumenBus, bem como seus requisitos e especificações. Ainda são apresentadas as ferramentas e tecnologias utilizadas no processo de implementação.

Finalmente, o capítulo 4 apresenta as conclusões deste trabalho. Além disso, são apresentadas as sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são discutidos os assuntos principais que são utilizados para a realização deste trabalho. Na seção 4.1 é realizada uma explanação sobre o tema transporte coletivo na cidade de Blumenau, principal foco deste projeto. Na seção 4.2 é realizada a definição de sistemas colaborativos. Na seção 4.3 é definido como ferramentas de comunicação são utilizadas para o apoio à colaboração. Por fim, a seção 4.4 apresenta os trabalhos correlatos ao presente trabalho.

2.1 TRANSPORTE COLETIVO NA CIDADE DE BLUMENAU

De acordo com informações publicadas no portal do Serviço Autônomo Municipal de Trânsito e Transportes de Blumenau (SETERB) (SETERB, 2017), o sistema de transporte coletivo da cidade conta com aproximadamente 90 linhas de ônibus que transitam entre seis terminais urbanos. Além desses, há outros dois terminais que estão sendo projetados (SETERB, 2017).

Após a troca para uma nova empresa licenciada para operar o serviço de transporte público de Blumenau, a contratada Blumob, realizou até o ano de 2017 a troca de 104 ônibus que foram substituídos por novos veículos 0km, correspondendo a cerca de 40% da frota (CAMARGO, 2017). Das 120 reclamações recebidas pela Ouvidoria do SETERB, em julho de 2017, apenas cinco dessas reclamações estão relacionadas à atrasos (CAMARGO, 2017).

O sistema de transporte integrado de Blumenau opera, principalmente, a partir de três tipos de linhas: troncais, que transportam os usuários diretamente entre os terminais por meio das principais vias da cidade; interbairros, que tem como objetivo realizar o transporte entre os terminais Fortaleza, Aterro, Proeb e Velha, sem passar pelo centro da cidade; e as alimentadoras, que favorecem ligações entre bairros e seus terminais próximos (SETERB, 2017).

Além destes, o serviço também oferece linhas entre o horário das 0h e 4h (chamadas linhas Madrugueiras), que operam entre os principais bairros da cidade. As linhas do trabalhador que atendem empresas e indústrias que possuem um elevado número de funcionários, possuem rotas definidas e operam em horários de maior demanda. Ainda, há os serviços pré-agendados, que visam atender pessoas portadoras de dificuldades severas de locomoção para o seu transporte até o local onde realizam fisioterapia (SETERB, 2017).

2.2 SISTEMAS COLABORATIVOS

Embora muitos prefiram utilizar o termo *groupware* para designar os sistemas computacionais que apoiam o trabalho em grupo e, separadamente o termo Computer Supported Cooperative Work (CSCW) para definir os sistemas e os efeitos psicológicos, sociais e organizacionais do trabalho em grupo, no Brasil o termo “Sistemas Colaborativos” foi adotado para designar ambos os termos (NICOLACI-DA-COSTA; PIMENTEL, 2012). O termo *groupware*, cunhado inicialmente em meados de 1978 por Peter e Trudy Johnson-Lenz, foi refinado uma década depois por Ellis como “sistema baseado em computador para dar suporte a grupos de pessoas engajadas numa tarefa comum (ou objetivo) e que provê uma interface para um ambiente compartilhado” (ELLIS et al., 1991).

O conhecimento não é um produto fixo e acabado, ele é construído num contexto de trocas, mediante um tensionamento constante entre o conhecimento atual (“certezas atuais”, provisórias) e as dúvidas que recaem sobre essas certezas, conduzindo ao estabelecimento de novas relações ou conhecimentos (novas certezas ainda que também temporárias) (FAGUNDES et al., 1999, apud CASTRO; MENEZES, 2012, p. 136).

Segundo Nicolaci-da-Costa e Pimentel (2012, p. 13), “cada sistema colaborativo constitui um ciberespaço específico”. Desta forma, as pessoas envolvidas no projeto e desenvolvimento de sistemas colaborativos criam por consequência novas formas de interação social e novos ambientes para este fim. Para os autores, “um sistema mais adequado à nova sociedade deve ser concebido para ser um espaço a ser habitado” (NICOLACI-DA-COSTA; PIMENTEL, 2012, p. 13). Os novos sistemas colaborativos devem focar nas necessidades de colaboração, interação e compartilhamento, de forma que tenham flexibilidade de horário e lugar, favorecendo a criação e a informalidade (NICOLACI-DACOSTA; PIMENTEL, 2012). Como demonstra a Figura 1, o modelo 3C de colaboração fundamenta os Sistemas Colaborativos.

Figura 1 - Modelo 3C de colaboração



Fonte: Ellis, Gibbs e Rein. (2006).

Segundo Vivacqua e Garcia (2012, p. 36), “A colaboração é um tema complexo, pois ocorre de formas diferentes e por motivos diversos.”. Como base, o Modelo 3C analisa a colaboração em três dimensões: comunicação, coordenação e cooperação (Figura 1). Caracteriza-se a comunicação como a troca de mensagens, a argumentação e a negociação entre indivíduos; a coordenação por sua vez é caracterizada pelo gerenciamento de pessoas, atividades e recursos; e a cooperação, por fim, é caracterizada pelo trabalho em conjunto em um espaço compartilhado, a fim de produzir objetos ou informações (FUKS et al., 2012). É importante notar que, apesar do modelo 3C separar a colaboração em dimensões com o intuito de focar nos aspectos mais relevantes para a sua análise, os “três C’s” se relacionam para que a colaboração ocorra (FUKS et al., 2012). Ainda, é importante destacar que é comum em um sistema colaborativo que um dos pilares seja mais evidenciado, conforme o intuito desse mesmo sistema.

2.3 FERRAMENTAS DE APOIO PARA SISTEMAS COLABORATIVOS

Conforme Pimentel, Gerosa e Fuks (2012, p. 66), “A internet se transformou num meio de comunicação humana de alcance mundial”. Devido a sua utilidade no contexto da interação humana, ferramentas de comunicação são frequentemente inseridas como apoio às aplicações colaborativas. Como consequência, a fim de atender às necessidades de diferentes contextos para sistemas colaborativos com focos específicos, tais ferramentas de apoio são constantemente aprimoradas e tem suas funcionalidades adaptadas. Como exemplo, o sistema de correio eletrônico que teve grande sucesso já nos primeiros anos de sua implantação. Posteriormente, a fim de possibilitar a comunicação entre vários indivíduos em simultâneo, o

mesmo sistema foi adaptado para a criação das listas de discussão (PIMENTEL; GEROSA; FUKS, 2012).

Em decorrência de um ambiente tecnológico em constante mutação e da necessidade de meios de comunicação que permitissem um ambiente de rápida produção e compartilhamento de informações, em meados de 2006 e 2007 surgiram redes sociais como o Twitter e o Tumblr (PIMENTEL; GEROSA; FUKS, 2012). Sistemas como estes trazem para o público a formalização dos conceitos de micro blog, que são caracterizados como adaptações dos blogs. Para atingir objetivos como agilidade na inserção de informações e informalidade, os micro blogs tem limite de caracteres para as postagens realizadas, sendo entre 140 e 200 caracteres (PIMENTEL; GEROSA; FUKS, 2012).

Com o tempo, os micro blogs foram incorporados também em outras redes sociais a fim de atender necessidades específicas da comunicação dos usuários. Devido à natureza sucinta das postagens deste tipo de sistema, existe de um lado a possibilidade da encadeação de comentários pertinentes à informação publicada e de outro a possibilidade de utilização da ferramenta como um novo meio de percepção para a tomada de decisão (PIMENTEL; GEROSA; FUKS, 2012). Segundo a visão de Pimentel, Gerosa e Fuks (2012, p. 84) sobre este tipo de sistema de comunicação, “A integração com outros sistemas e sites, bem como o uso diversificado em áreas como jornalismo e educação, têm sido algumas das tendências observadas na evolução dos micro blogs.”.

2.4 TRABALHOS CORRELATOS

Foram selecionados três trabalhos correlatos com objetivos semelhantes aos definidos neste trabalho. Todos foram planejados para atender a necessidade de acesso a uma plataforma que disponibiliza informações essenciais para a utilização do transporte coletivo. A seção 2.4.1 detalha o aplicativo BluBus lançado em 2017 na cidade de Blumenau, Santa Catarina (SC). A seção 2.4.2 detalha o aplicativo MovelBus, que apesar de também ter foco em locais específicos, opera com as linhas de transporte coletivo de três cidades diferentes. Por fim, a seção 2.4.3 detalha o aplicativo BusMaps, resultado de um trabalho de conclusão de curso aplicado para o sistema de transporte coletivo da cidade de Florianópolis/SC.

2.4.1 BluBus




O aplicativo BluBus tem como objetivo disponibilizar uma fonte de consulta aos horários do transporte coletivo da cidade de Blumenau/SC. A aplicação móvel foi

desenvolvida buscando a simplicidade de uso, praticidade, sendo que não é necessário o acesso à internet (JENSEN, 2017).

Para efetuar a consulta dos horários disponíveis é possível selecionar o ponto de saída, a linha desejada e até mesmo o turno do ônibus. Para que o acesso às informações se torne mais rápido, existe a possibilidade do usuário salvar suas linhas de transporte favoritas. O aplicativo também conta com um sistema de notificação que alerta o usuário quando o horário desejado está próximo. Trata-se de uma aplicação gratuita, sem intenção de lucro, que visa auxiliar a população de Blumenau/SC em seu dia-a-dia (JENSEN, 2017).

O BluBus possui uma interface inicial na qual é possível listar os pontos de saída e as linhas de ônibus para seleção e, se necessário, refinar esta lista a partir de uma entrada do usuário no campo de busca (Figura 2 (a)). A partir daí são disponibilizados em forma de grade os horários da linha de ônibus de acordo com as opções selecionadas pelo usuário, e na mesma interface pode ser alterado o turno pelo qual se deseja consultar: dias úteis, sábados e domingos ou feriados (Figura 2 (b)). Não foi encontrado material referente ao ferramental utilizado para o desenvolvimento do aplicativo.

Figura 2 - Seleção para pesquisa

a)	b)		
  	Dias Úteis Sábados Dom/Fer		
SELECIONE A SAÍDA SELECIONE A LINHA Refina a busca aqui	TERMINAL ATERRO Todos os horários 12 TRONCAL - VIA ESCOLA AGRICOLA Rota: Terminal Aterro/ Terminal Proeb/ Terminal Fonte Próximo ônibus: 19:04		
ALVORADA	03:55	10:15	16:32
ANDRE NICOLETTI	04:30	10:30	16:43
ARARANGUA	04:45	10:45	16:54
BADENFURT	05:00	11:00	17:05
BADENFURT	05:15	11:10	17:14
BADENFURT	05:30	11:20	17:22
BELO HORIZONTE	05:41	11:30	17:30
BOA VISTA	05:52	11:35 *	17:30
	06:02	11:35 *	17:30

Fonte: elaborado pelo autor.

2.4.2 MoveIBus

Assim como outras ferramentas de consulta de informações com ênfase local, o MoveIBus é um aplicativo gratuito. O objetivo desta aplicação também é disponibilizar ao

usuário todos os horários de ônibus cadastrados em um banco de dados interno. Assim como o BluBus, o MoveBus possibilita uma consulta rápida a qualquer momento sem necessidade de acesso à Internet (MOVELBUS, 2017). O aplicativo pode ser instalado em celulares mais simples e compatíveis com Java a partir de uma versão Wireless Application Protocol (WAP), pelo qual é realizado o acesso ao portal via navegador, ou smartphones com Android (MOVELBUS, 2017). O MoveBus, apesar de contar com uma base de dados defasada, possibilita a consulta de linhas e horários de ônibus das cidades catarinenses de Blumenau e Florianópolis, e também da cidade de Porto Alegre, Rio Grande do Sul (RS).

Na listagem de horários, após a consulta inicial a partir da cidade e linha selecionada, também existem ressalvas referentes ao trajeto realizado pelo ônibus, caso aquele horário em questão possua algum desvio de percurso. Essa situação é ilustrada na Figura 3, na linha de número 702, bairro Vila Nova. Os horários de 06:45, 09:05, 10:35 e 11:30 realizam uma rota diferenciada dos demais. Não foi encontrado material referente ao ferramental utilizado para o desenvolvimento do aplicativo.

Figura 3 - Horários com observação de desvio

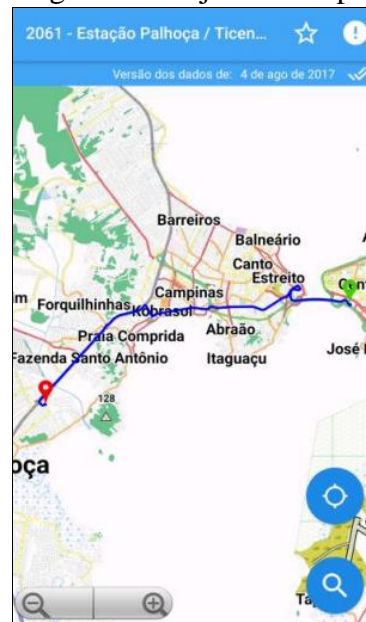


Fonte: elaborado pelo autor.

2.4.3 BusMaps

O BusMaps é um aplicativo desenvolvido para a plataforma Android, que visa facilitar a consulta aos horários das linhas de ônibus da cidade de Florianópolis/SC. A aplicação possui como funcionalidade, além da consulta básica de informações referentes aos horários das linhas de ônibus da cidade, a disponibilização de mapas e trajetos utilizados pelos ônibus, conforme ilustrado pela Figura 4 (MOREIRA, 2013).

Figura 4 - Trajeto no mapa



Fonte: elaborado pelo autor.

Para tal, Moreira (2013) realizou a coleta e adequação dos dados das linhas de ônibus da página da Prefeitura Municipal de Florianópolis. Para a utilização das informações na ferramenta desenvolvida, foi realizada uma avaliação e correção de inconsistências nos arquivos de trajetos coletados. Também foi proposta a utilização de mapas do OpenStreetMap substituindo outras opções comerciais como as oferecidas pelo Google e Bing. Apesar do foco inicial ser o desenvolvimento do aplicativo para a finalidade da consulta de informações dos ônibus de Florianópolis/SC, é sugerida a criação de uma ferramenta que futuramente possa ser generalizada para utilização em outras cidades (MOREIRA, 2013).

3 DESENVOLVIMENTO

Neste capítulo é apresentado o detalhamento do desenvolvimento do aplicativo e suas especificações. A seção 3.1 apresenta um levantamento das informações. A seção 3.2 apresenta a especificação, além de enumerar os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) do aplicativo. A seção 3.3 demonstra as técnicas e ferramentas utilizadas no desenvolvimento do trabalho, assim como trechos de códigos-fontes relevantes e sua operacionalidade. Por fim, a seção 3.4 apresenta os resultados e discussões obtidos com a realização desse trabalho.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta a construção de um aplicativo móvel, que visa disponibilizar informações referentes ao transporte coletivo da cidade de Blumenau, de forma que também exista colaboração entre os usuários de uma mesma linha de ônibus. Para que seja possível consultar informações e colaborar, é necessário que o usuário esteja autenticado. No aplicativo é possível realizar a consulta dos horários de saída de ônibus de acordo com o número da linha, terminal de saída e dia da semana. Também existe a possibilidade de que o usuário consulte um mapa, onde são marcados os pontos de ônibus pelos quais a linha passa durante o trajeto. Além destas informações básicas, é possível que sejam feitos comentários sobre uma linha, de maneira que possa haver disseminação de informação entre os usuários do aplicativo. Para o usuário que realiza consultas frequentes à uma linha específica, existe uma opção para marca-la como favorita, e desta forma, a mesma será apresentada em uma lista de favoritos na tela inicial do aplicativo.

O trabalho desenvolvido utiliza como módulo servidor o Firebase, serviço disponibilizado pela Google, para consumo da base de dados (Firebase Realtime Database) e também para serviço de autenticação (Firebase Authentication). Para o módulo cliente, optou-se por construir a interface através da utilização do *framework* Ionic, construída sobre a plataforma Cordova que permite o desenvolvimento de aplicações híbridas sem necessidade de implementação diretamente em Android ou iOS.

3.2 ESPECIFICAÇÃO

Esta seção apresenta a especificação do sistema desenvolvido, através dos requisitos funcionais e não funcionais, e também dos diagramas apresentados nas subseções. O Quadro 1 apresenta os Requisitos Funcionais (RF) do aplicativo desenvolvido junto com a matriz de rastreabilidade dos casos de uso e o Quadro 2 lista os Requisitos Não Funcionais (RNF). Na

seção 3.2.1 é apresentado o diagrama de casos de uso. Na seção 3.2.2 é apresentada a estruturação do banco não relacional para a aplicação desenvolvida.

Quadro 1 - Requisitos Funcionais e a matriz de rastreabilidade dos UCs

Requisitos funcionais	Caso de Uso
RF01: O sistema deve permitir que o usuário realize consulta aos horários do ônibus de acordo com a linha, terminal de saída e dia selecionados.	UC03
RF02: O sistema deve permitir que o usuário pesquise pelo número ou descrição da linha que deseja, a fim de filtrar a lista de linhas disponíveis.	UC01
RF03: O sistema deve permitir que o usuário visualize o trajeto realizado pelo ônibus, de acordo com a linha e terminal de saída selecionados.	UC06
RF04: O sistema deve exibir o trajeto da linha de ônibus em um mapa, de acordo com o terminal de saída selecionado.	UC06
RF05: O sistema deve exibir o trajeto na forma de marcações nos pontos de ônibus pelos quais a linha passa e para durante sua rota.	UC06
RF06: O sistema deve permitir que o usuário realize a inserção de comentário em uma linha selecionada.	UC05
RF07: O sistema deve permitir que o usuário visualize os comentários em uma linha selecionada.	UC04
RF08: O sistema deve permitir que o usuário remova seu comentário.	UC05
RF09: O sistema deve permitir que o usuário salve uma linha de ônibus como favorita.	UC02

Fonte: elaborado pelo autor.

Quadro 2 - Requisitos Não Funcionais

Requisitos não funcionais
RNF01: O aplicativo deve aplicar técnicas de Sistemas Colaborativos para a captura e geração de informação de forma colaborativa.
RNF02: O <i>frontend</i> deverá ser implementada na linguagem de programação Javascript.
RNF03: O aplicativo móvel deverá ser construído com o <i>framework</i> Ionic.
RNF04: Deverá utilizar as normas de usabilidade voltadas a aplicações móveis.

Fonte: elaborado pelo autor.

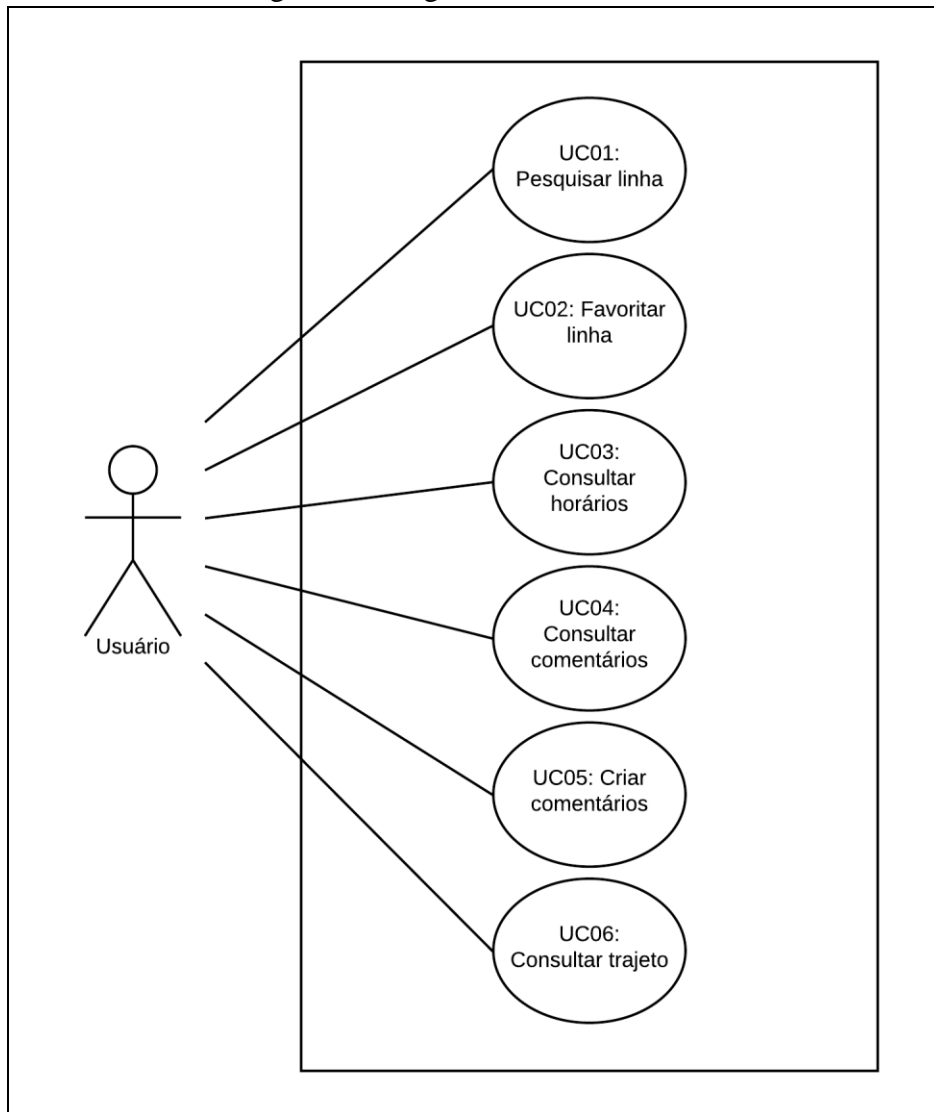
3.2.1 Diagrama de Casos de uso

A Figura 5 apresenta o diagrama de casos de uso do aplicativo móvel. Neste diagrama é representado o único ator do trabalho, o usuário, que realiza todas as interações possíveis com o sistema.

O ator usuário representa o usuário final que utiliza o transporte público e deseja realizar a consulta das informações do ônibus através do aplicativo, de acordo com sua necessidade. Este ator pode realizar atividades como consultar linhas, horários, comentários

referentes a uma linha específica e trajetos. Também possui o privilégio de criar novos comentários para uma linha de ônibus.

Figura 5 - Diagrama de casos de uso



Fonte: elaborado pelo autor.

Conforme a figura demonstra, os casos de uso são pesquisar linha, favoritar linha, consultar horários, consultar comentários, criar comentários e consultar trajeto. O caso de uso pesquisar linha descreve os procedimentos a serem realizados para que o ator usuário busque por uma linha de ônibus da qual necessita consultar informações. Esta pesquisa pode ser realizada manualmente através da lista completa disponibilizada na tela de linhas, ou através do campo de pesquisa sobre esta lista, em que o ator usuário pode digitar o número da linha ou sua descrição e o aplicativo fará o filtro sobre a lista completa de linhas de acordo com o que foi informado pelo ator usuário no campo de pesquisa.

O caso de uso `favoritar linha` descreve os procedimentos realizados pelo ator `usuário` para armazenar uma determinada linha de ônibus em sua lista de favoritos, que é apresentada na tela principal do aplicativo. Esta ação é realizada através da tela de informações da linha de ônibus, em que o ator `usuário` deve tocar sobre o botão com o ícone de estrela. Caso a linha exibida em tela ainda não esteja na lista de favoritos do ator `usuário`, o ícone no botão será apresentado apenas com o contorno da figura de uma estrela, sem preenchimento. Ao tocar sobre o botão, este ícone será preenchido com cor e será apresentada uma mensagem ao ator `usuário` indicando que aquela linha foi adicionada à sua lista de favoritos. Caso a linha exibida em tela já esteja na lista de favoritos do ator `usuário`, o ícone no botão será apresentado como a figura de uma estrela preenchida com cor. Ao tocar sobre o botão, o seu ícone terá seu preenchimento removido, sendo mostrado apenas como o contorno da figura de uma estrela e uma mensagem será apresentada ao ator `usuário` indicando que aquela linha foi removida de sua lista de favoritos.

O caso de uso `consultar horários` descreve o processo a ser realizado pelo ator `usuário` para que o mesmo possa visualizar os horários de partida de uma determinada linha de ônibus, de acordo com o ponto de saída da linha e com o dia da semana (dias úteis, sábados e domingos e feriados). A partir da tela de informações da linha, o ator `usuário` deve selecionar um ponto de saída e um dia da semana nos respectivos campos, e então tocar sobre o botão `Filtrar`. Após isto, o aplicativo carregará em tela uma lista de horários da linha de ônibus, de acordo com as informações preenchidas pelo ator `usuário` nos campos de filtro.

O caso de uso `consultar comentários` descreve o processo realizado pelo ator `usuário` para que o mesmo possa visualizar comentários feitos por outros usuários do aplicativo, referentes à uma determinada linha de ônibus. A partir da tela de informações da linha, o ator `usuário` deve tocar sobre o botão com o ícone de um balão de fala e desta maneira será redirecionado para a tela de comentários.

O caso de uso `criar comentários` descreve o processo a ser realizado pelo ator `usuário` para que o mesmo possa fazer um ou mais comentários sobre uma determinada linha de ônibus. A partir da tela de comentários, o ator `usuário` deve digitar no campo de texto o que deseja comentar sobre a linha de ônibus, e então confirmar a inclusão. Após isto, a tela de comentários será atualizada e mostrará na lista o comentário realizado pelo ator `usuário`, junto com os demais comentários realizados por ele ou outros usuários. O ator `usuário` também pode remover o comentário que realizou, tocando sobre o ícone de três pontos verticais que fica ao lado de cada comentário de sua autoria. Então, será apresentada em tela a

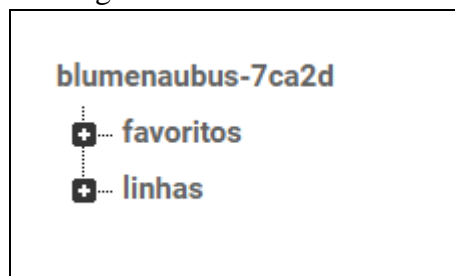
opção *Excluir*, na qual o ator *usuário* deverá tocar para remover seu comentário. Após isto, a tela de comentários será atualizada e não será mais possível para nenhum usuário da aplicação visualizar aquele comentário.

O caso de uso *consultar trajeto* descreve o processo a ser realizado pelo ator *usuário* para que o mesmo possa visualizar o trajeto de uma determinada linha de ônibus, através da marcação dos pontos de ônibus pelos quais aquela linha realiza parada, em um mapa. Após realizar o filtro dos horários na tela de informações da linha, o ator *usuário* deve tocar sobre um horário da lista e então será redirecionado para a tela do mapa. Nesta tela, o ator *usuário* poderá visualizar no mapa as marcações dos pontos de ônibus através de círculos vermelhos que ficam sobre o mesmo. Através da visualização das marcações sobre os pontos de ônibus pelos quais a linha realiza parada, o ator *usuário* pode visualizar o trajeto realizado pela linha de ônibus até seu ponto de destino.

3.2.2 Estrutura do banco de dados não relacional

Nesta seção é apresentada a estrutura criada no banco de dados não relacional, para disponibilização das informações necessárias ao aplicativo, e uma breve explicação da estruturação. A Figura 6 demonstra os nós principais da estrutura, composta por *favoritos* e *linhas*.

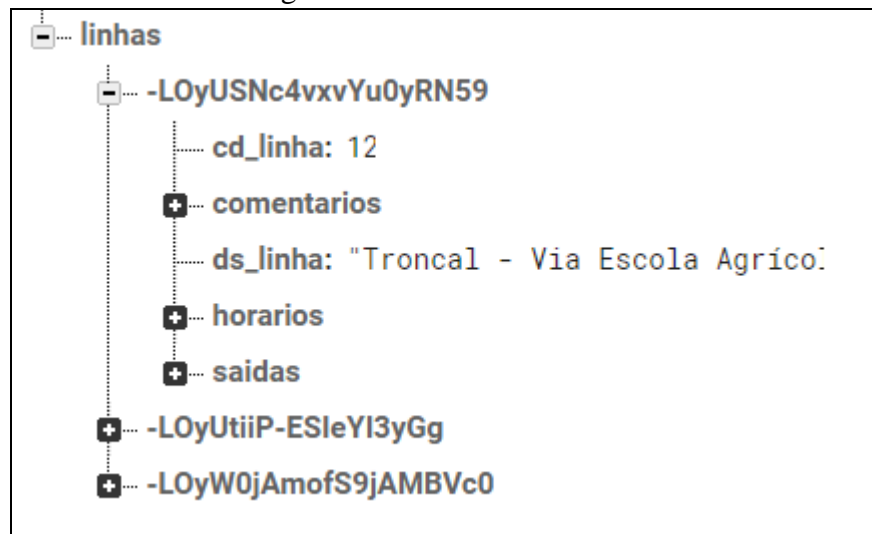
Figura 6 - Estrutura inicial



Fonte: elaborado pelo autor.

Dentro do nó *favoritos*, ficam armazenadas as informações referentes a cada linha de ônibus marcada como favorita por um usuário. Este nó possui filhos criados de acordo com a demanda, identificados por chaves únicas geradas dinamicamente. Cada filho possui as seguintes informações: código da linha, descrição da linha e ID do usuário. Dentro do nó *linhas*, ficam armazenadas todas as linhas cadastradas no banco de dados e suas respectivas informações. Como é possível visualizar na Figura 7, dentro de cada filho do nó *linhas* existem outros nós que guardam novas estruturas necessárias para armazenar as informações de cada linha de ônibus.

Figura 7 - Estrutura das linhas



Fonte: elaborado pelo autor.

Cada linha dentro do nó `linhas` é identificada por uma chave única gerada dinamicamente no momento de sua inserção no banco de dados. Toda linha possui um código e uma descrição e também as subestruturas necessárias para armazenar os comentários, horários e saídas da linha. Cada comentário dentro do nó `comentarios` é identificado por uma chave única e possui informações como: texto do comentário, data do comentário e e-mail, ID e nome do usuário criador. O nó `horarios` é dividido internamente de acordo com as datas (dias úteis, sábados ou domingos e feriados), e dentro destes existe uma divisão de acordo com o terminal de saída. Todo horário dentro desta estrutura também possui um identificador único e armazena a informação da hora de partida da linha. Por sua vez, o nó `saidas` armazena cada terminal de saída possível para aquela linha de ônibus, em que ficam guardadas as coordenadas daquele terminal, sua descrição e um novo nó contendo os dados de todos os pontos de ônibus pelos quais aquela linha passará e fará parada durante o seu trajeto.

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas para o desenvolvimento do aplicativo, assim como a sua operacionalidade. Também é descrita a implementação do aplicativo, apresentando trechos de códigos-fontes da aplicação desenvolvida. São apresentados detalhes do *frontend* e *backend* da aplicação, do banco de dados não relacional e serviço de autenticação.

3.3.1 Técnicas e ferramentas utilizadas

Esta subseção apresenta as técnicas e ferramentas utilizadas para a construção do aplicativo. Para o desenvolvimento deste trabalho foram utilizadas as ferramentas:

- a) Visual Studio Code editor de código;
- b) Firebase como *backend*;
- c) Firebase Realtime Database como banco de dados;
- d) Console do Firebase para manipular a estrutura do banco de dados;
- e) Postman para manipular o banco de dados e como ferramenta para testes das requisições HTTP;
- f) Firebase Authentication como serviço de autenticação de usuários;
- g) Node.js como plataforma de desenvolvimento;
- h) Javascript como linguagem de programação *frontend*;
- i) TypeScript como superset para a linguagem Javascript;
- j) Angular como plataforma e *framework* para o desenvolvimento *frontend*;
- k) Cordova como *framework* para desenvolvimento do aplicativo;
- l) Ionic como biblioteca para desenvolvimento das telas do aplicativo;
- m) Leaflet como *framework* para os recursos de mapeamento;
- n) OpenStreetMap como *layer* do mapa;
- o) SASS como pré-processador de CSS;
- p) Lucidchart como ferramenta para criação dos diagramas.

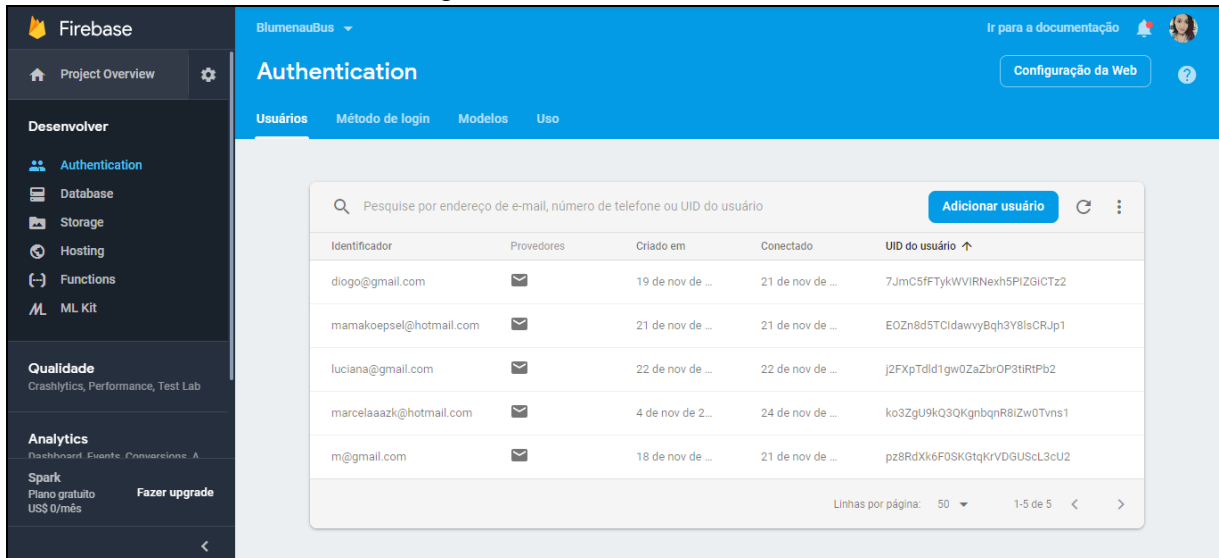
O *frontend* do aplicativo foi desenvolvido de forma responsiva sobre a plataforma Cordova, utilizando os recursos da biblioteca Ionic para criação das interfaces, o que possibilita a criação de aplicações híbridas, sem a necessidade de implementação em linguagem específica para Android ou iOS. Desta forma, foi utilizado o TypeScript para a escrita do código-fonte, com a linguagem de programação Javascript para desenvolvimento do *frontend*, e o *framework* Angular.

Para realização da marcação das rotas dos ônibus através de mapa, foi utilizado o *framework* Leaflet, utilizando como *layer* de mapa o OpenStreetMap. O OpenStreetMap é um projeto de mapeamento colaborativo que disponibiliza as imagens de mapas de forma gratuita e sem necessidade de cadastro ou criação de *token* para utilização dentro do desenvolvimento de aplicações de terceiros.

Para módulo servidor, foi utilizado o serviço Firebase da Google, que disponibiliza *backend* como um serviço para consumo. Desta forma, não há necessidade da criação e implementação do *backend* específico para o aplicativo desenvolvido neste trabalho. Isto se torna possível, pois este aplicativo não demanda regras de negócio complexas a serem tratadas no lado do servidor e utiliza o módulo apenas para o consumo de dados e

autenticação dos usuários. Dentro deste serviço, foi utilizado como banco de dados o Firebase Realtime Database, um banco não relacional no qual os dados são sincronizados em tempo real e armazenados como JSON e hospedados na nuvem. Como serviço de autenticação de usuários foi utilizado o Firebase Authentication, que salva em seu próprio repositório os dados dos usuários criados. Através do console do Firebase é possível realizar a administração tanto do serviço de autenticação de usuários quanto do banco de dados (Figura 8).

Figura 8 - Console do Firebase



The screenshot shows the Firebase Authentication console for the project 'BlumenauBus'. The main content area displays a table of users with the following data:

Identificador	Provedores	Criado em	Conectado	UID do usuário ↑
diogo@gmail.com	✉	19 de nov de ...	21 de nov de ...	7JmC5fTykVVIRNexh5PIZGICTz2
mamakoepsel@hotmail.com	✉	21 de nov de ...	21 de nov de ...	EO2n8d5TCIdawvyBqh3Y8lsCRJp1
luciana@gmail.com	✉	22 de nov de ...	22 de nov de ...	j2FXpTdlid1gw0ZaZbrOP3tIRIPb2
marcelaaazk@hotmail.com	✉	4 de nov de 2...	24 de nov de ...	ko3ZgU9KQ3QKgnbqnR8I2w0Tvns1
m@gmail.com	✉	18 de nov de ...	21 de nov de ...	pz8RdXk6F0SKGtqKrVDGUScL3cU2

The interface includes a search bar at the top with the text 'Pesquise por endereço de e-mail, número de telefone ou UID do usuário' and a button 'Adicionar usuário'. The table has a pagination control at the bottom showing 'Linhas por página: 50' and '1-5 de 5'.

Fonte: elaborado pelo autor.

3.3.2 Desenvolvimento da aplicação

Nesta subseção é apresentado o processo de desenvolvimento do aplicativo, ilustrado com os principais trechos de códigos-fontes da implementação, utilizando exemplos do trabalho desenvolvido.

3.3.2.1 Criação da aplicação móvel

A construção da aplicação foi realizada utilizando o *framework* Ionic que trabalha sobre a plataforma Cordova, além do *framework* Angular. O Cordova possibilita que o desenvolvedor implemente a aplicação através da linguagem Javascript, neste caso também utilizando o Angular, escrevendo a aplicação em códigos reativos para a interface web que posteriormente são compilados para a criação do aplicativo móvel. Através do Cordova é possível realizar *build* da aplicação web para plataforma Android ou iOS. O Ionic por sua vez, diz respeito a construção da parte visual da aplicação e do controle de como esta interface será apresentada para o usuário final.

A estrutura da aplicação é criada através dos *templates* HTML, *styles* CSS e scripts TypeScript. O HTML é responsável pela estrutura da interface com o usuário, é estilizado através das regras cadastradas no arquivo CSS e seu comportamento é controlado através dos scripts, da mesma maneira que em aplicações web comuns. O Ionic possui componentes específicos para a criação de aplicativos móveis, os quais são inseridos na estrutura HTML como componentes, da mesma forma que qualquer componente HTML. O Ionic possui *tags* de cabeçalho, corpo e rodapé próprios para criação da interface do aplicativo, que são chamados de `ion-header`, `ion-content` e `ion-footer` respectivamente.

No Quadro 3 é ilustrado o HTML da tela do mapa, em que se pode visualizar nas linhas 1 e 3 a utilização do componente de cabeçalho próprio do Ionic e nas linhas 5 e 10 o componente de corpo da estrutura HTML, do Ionic. Além dos componentes base de uma estrutura HTML, o Ionic também possui uma variedade de componentes próprios em sua biblioteca.

Quadro 3 - HTML da tela do mapa

```

1 <ion-header>
2   <ion-navbar> <ion-title> </ion-title> </ion-navbar>
3 </ion-header>
4
5 <ion-content>
6   <div class="header-linha-selec">
7     <h4>{{ dsLinha }}</h4>
8   </div>
9   <div id="map" style="width:100%; height:100%;"></div>
10 </ion-content>
11

```

Fonte: elaborado pelo autor.

Conforme mostrado no Quadro 4, que ilustra parte do HTML da tela de comentários, existem diferentes componentes que fazem o trabalho de renderizar o HTML. Em alguns componentes, como na linha 12, são utilizadas diretivas `*ngIf` que realizam a função de validar condições `se`. Estas condições controlam se o componente deverá ser mostrado em tela ou não, dependendo da validação inserida na diretiva.

Quadro 4 - HTML da tela de comentários

```

1 <ion-header>
2 | <ion-navbar> <ion-title></ion-title> </ion-navbar>
3 </ion-header>
4
5 <ion-content padding>
6 | <iframe name="hiddenFrame" class="hide"></iframe>
7 | <div padding-top class="header-linha-selec">
8 | | <h4>{{ dsLinha }}</h4>
9 | </div>
10 <ion-card *ngFor="let coment of comentarios">
11 | <button class="colunaBtn" ion-button icon-only clear small float-right
12 | *ngIf="coment.uid == uid" on-click="presentActionSheet(coment.key)">
13 | | <ion-icon name="more"></ion-icon>
14 | </button>
15 | <ion-card-content>
16 | | <div class="areaComent">
17 | | | <div class="mensagem">
18 | | | | <p>{{ coment.ds_comentario }}</p>
19 | | | </div>
20 | | </div>
21 | </ion-card-content>
22
23 | <ion-row class="footComentario">
24 | | <ion-col class="colunaInfo">
25 | | | <ion-item>
26 | | | | <p>{{ coment.username }} ({{ coment.email }})</p>
27 | | | | <p>{{ coment.dt_comentario }}</p>
28 | | | </ion-item>
29 | | </ion-col>
30 | </ion-row>
31
32 </ion-card>
33 </ion-content>

```

Fonte: elaborado pelo autor.

Também é possível definir ações para determinados elementos HTML. Na linha 12 do mesmo quadro é realizada a atribuição de uma ação ao evento de toque sobre um elemento HTML. Neste caso, um botão fazendo com que esse elemento dispare a ação `presentActionSheet` (localizada no script que realiza o controle da tela) quando o usuário interagir com o mesmo utilizando o toque da tela no dispositivo móvel, sobre o referido botão.

Os scripts de controle são arquivos TypeScript responsáveis por realizar o processamento da aplicação no dispositivo cliente, como carregar informações armazenadas no banco, realizar a validação e comparação de dados e também a comunicação com o serviço do Firebase. Neste script podem ser declarados atributos e propriedades que o `template` é capaz de acessar através das definições declaradas nos componentes da estrutura HTML. No início do script TypeScript é necessário realizar a importação de componentes, páginas e demais serviços que serão necessários para o desenvolvimento do fluxo daquela tela. Após isso, através da anotação `Component`, identifica-se que este arquivo será um componente da

aplicação. Dentro desta anotação deve-se fazer referência ao arquivo de estrutura HTML que será utilizado como `template` do componente. Seguindo a estrutura do script, é iniciada a declaração da classe em `si`, que representa o controle daquele componente. Pode-se declarar atributos e propriedades de acordo com a necessidade. Foi criado o construtor da classe que será iniciado na criação do componente em tela e demais métodos necessários para o desenvolvimento do fluxo daquela tela.

No Quadro 5 é demonstrado parte do código-fonte da tela de informações da linha, que ilustra a importação das dependências do script desde a linha 1 até a linha 6. Logo após, são listados os atributos utilizados na tela, contidos entre a linha 14 e a linha 24. Estes atributos são utilizados no `template` dessa mesma tela ou nos métodos criados para manipular as rotinas. Após a declaração dos atributos, é implementado o construtor da classe. As ações implementadas dentro da declaração do construtor serão realizadas logo que o componente for renderizado em tela.

Quadro 5 - Script da tela de informações da linha

```

1  import { Component, ViewChild, ElementRef } from '@angular/core';
2  import { NavController } from 'ionic-angular';
3  import { NavParams } from 'ionic-angular';
4  import { snapshotToArray } from '../linhas/linhas';
5  import leaflet from 'leaflet';
6  import * as firebase from 'firebase';
7
8  @Component({
9    selector: 'page-mapa',
10   templateUrl: 'mapa.html'
11 })
12 export class Mapa {
13   @ViewChild('map')
14   mapContainer: ElementRef;
15   map: any;
16   pontos = [];
17   ref = null;
18   linha = null;
19   cdLinha = null;
20   dsLinha = null;
21   dsSaida = null;
22   saida = null;
23   latSaida = null;
24   lngSaida = null;
25
26   constructor(public navCtrl: NavController, NavParams: NavParams) {
27     this.linha = NavParams.data.linha;
28     this.cdLinha = this.linha.cd_linha;
29     this.dsLinha = this.linha.ds_linha;
30
31     this.dsSaida = NavParams.data.saida;
32     this.ref = firebase.database().ref('linhas/').child(this.linha.key).child('saidas').orderByChild('ds_saida');
33     this.ref.equalTo(this.dsSaida).once('value', resp => {
34       this.saida = snapshotToArray(resp)[0];
35     });
36   }
37
38   ionViewDidEnter() {
39     this.loadmap();
40   }

```

Fonte: elaborado pelo autor.

O estilo das páginas HTML é definido através de CSS. Para definição da cor de fundo das telas, da cor da fonte dos textos e da cor primária dos componentes utilizados, são utilizadas variáveis através do SASS, para que seja possível facilitar a aplicação destas informações de estilo de forma geral na aplicação. Para estilizações específicas de cada tela e de cada componente, são utilizadas as estruturas padrões comuns dos arquivos CSS, que nesta aplicação devido a utilização do SASS são arquivos de extensão `.scss`. O Quadro 6 demonstra a utilização de variáveis para estilização geral da aplicação, através da aplicação do SASS. Na linha 13 a variável `$text-color` recebe a cor que deve ser aplicada sobre a fonte dos textos da aplicação, na linha 14 a variável `$background-color` define a cor que é utilizada no fundo de todas as telas e na linha 16 a variável `$colors` recebe todas as cores que devem ser aplicadas nos componentes.

Quadro 6 - Variáveis SASS

```

12
13 $text-color:      ■ #000;
14 $background-color: □ #fff;
15
16 $colors: (
17   primary:      ■ #504e4e,
18   secondary:    ■ #32db64,
19   danger:       ■ #f53d3d,
20   light:        □ #f4f4f4,
21   dark:         ■ #222,
22   favorite:     ■ #698878
23 );
24

```

Fonte: elaborado pelo autor.

A disponibilização do trajeto da linha de ônibus é realizada através de um mapa, onde são marcados os pontos de ônibus pelos quais a linha realizará paradas. Para tal, foi optado por utilizar o OpenStreetMap como *layer* de mapa através da biblioteca Leaflet. A localização dos pontos de parada é realizada através do banco de dados, em que são cadastradas suas respectivas coordenadas. Estes são marcados no mapa através de círculos vermelhos sobre os pontos de ônibus de passagem e parada da linha, utilizando o Leaflet.

O Quadro 7 mostra o método `loadMap`, responsável pela inicialização do mapa em tela e marcação dos pontos de ônibus de acordo com a linha e ponto de saída selecionados na tela de informações da linha. Os atributos `latSaida` (linha 49) e `lngSaida` (linha 50) são consumidos da estrutura no banco de dados que armazena os possíveis pontos de saída inicial da linha e a partir destes é realizada a centralização do foco de visualização do mapa (linha 65). Iniciando na linha 58, é carregado o mapa através da biblioteca Leaflet utilizando o endereço do *layer* disponibilizado pelo OpenStreetMap. O atributo `pontos` (linha 52)

armazena as coordenadas de todos os pontos de ônibus pelos quais a linha passará (linha 54). Através destes dados, é iniciado um loop sobre o atributo pontos (linha 67), que processará a criação e armazenamento no atributo circle (linha 69), de marcações em forma de círculos na cor vermelha (linha 70), e posteriormente adiciona cada marcação no mapa (linha 74).

Quadro 7 - Método de inicialização do mapa

```

48 loadmap() {
49   this.latSaida = this.saida.cordSaida.lat;
50   this.lngSaida = this.saida.cordSaida.lng;
51
52   this.pontos = [];
53   if (this.saida.pontos) {
54     this.pontos = Object.keys(this.saida.pontos).map(key => this.saida.pontos[key]);
55   }
56
57   this.map = leaflet.map('map').fitWorld();
58   leaflet.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
59     attributions:
60       'Map data &copy; <a href="http://openstreetmap.org">OpenStreetMap</a> /n' +
61       'contributors, <a href="http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a> /n' +
62       ', Imagery © <a href="http://mapbox.com">Mapbox</a>',
63     maxZoom: 18
64   }).addTo(this.map);
65   this.map.setView(new leaflet.LatLng(this.latSaida, this.lngSaida), 15);
66
67   for (let pt in this.pontos) {
68     if (this.pontos[pt].lat && this.pontos[pt].lng) {
69       let circle = leaflet.circle([this.pontos[pt].lat, this.pontos[pt].lng], {
70         color: 'red',
71         fillColor: '#f03',
72         fillOpacity: 0.5,
73         radius: 12
74       }).addTo(this.map);
75     }
76   }
77 }
78 }
79

```

Fonte: elaborado pelo autor.

A partir de estruturas como as mostradas, é desenvolvida uma aplicação web padrão que pode ser rodada através de navegadores. Para que seja possível criar a aplicação móvel através da estrutura desenvolvida, é necessário utilizar a plataforma Cordova que é usada como base do *framework* Ionic para compilar a aplicação. Este processo encapsulará a aplicação web em um aplicativo móvel compatível com a plataforma desejada (Android ou iOS). Se a aplicação for compilada para o sistema operacional Android, é necessário possuir o Android SDK na máquina que estiver realizando este processo. No aplicativo desenvolvido neste trabalho, foi realizada a compilação apenas para a plataforma Android, de acordo com os dispositivos disponíveis para instalação e testes.

3.3.2.2 Estruturação do serviço de *backend*

Para o desenvolvimento da aplicação foi utilizado o Firebase, serviço da Google, como serviço de *backend*. O Firebase é reconhecido como um *backend as a service* (BAAS), ou seja, serve como um *middleware* para facilitar a conexão do *frontend* de uma aplicação móvel ou web com serviços em nuvem através de API e SDK. O Firebase também conta com o Firebase Realtime Database, que é um serviço de banco de dados não relacional hospedado na nuvem e o Firebase Authentication, que é um provedor de serviço de autenticação, além de outros serviços que não foram utilizados no desenvolvimento deste aplicativo.

Antes de poder realizar qualquer conexão com os serviços que o Firebase possui, é necessário gerar uma série de *tokens* através do painel de controle do mesmo, que devem ser identificados no *frontend* da aplicação. Caso não exista a identificação destes *tokens* dentro do aplicativo, o mesmo não conseguirá estabelecer conexão com os serviços do Firebase.

Como serviço de banco de dados, foi utilizado o Firebase Realtime Database. Através da API do Firebase, são utilizadas funções disponibilizadas para realização da conexão sincronizada ou não com a base criada no serviço da Google. Em alguns momentos na implantação deste aplicativo, foi optado por utilizar a conexão sincronizada, para que as informações apresentadas em tela se mantivessem atualizados caso houvesse qualquer inserção, alteração ou exclusão de registros naquela referência da base de dados. Como é ilustrado no Quadro 8, a conexão sincronizada com o Firebase Realtime Database se dá através do método `on` chamado sobre a referência do banco para qual se quer apontar. Na linha 39 foi necessário ligar a sincronização com a referência `favoritos`. Quando o usuário marcar como favorito ou desmarcar uma linha, e retornar para a página inicial, automaticamente a linha será adicionada à lista ou removida, sem necessidade de reativação da tela via programação. Na linha 43 realiza-se uma chamada para a referência `linhas`, porém sem ativar a sincronização automática com a base de dados, através do método `once`. Para que seja possível utilizar a API do Firebase para realizar as conexões com a base de dados, é necessário importar o módulo do Firebase, conforme linha 17. Através deste módulo realiza-se as chamadas para os serviços oferecidos pelo Firebase, conforme necessidade.

Quadro 8 - Conexão com a base de dados

```

16
17 import * as firebase from 'Firebase';
18
19 @Component({
20   templateUrl: 'home.html',
21   providers: []
22 })
23 export class HomePage {
24
25   private nav: Nav;
26
27   favoritos = [];
28   ref = firebase.database().ref('favoritos/');
29
30   btnSegment = 'favoritos';
31
32   linhas = [];
33   linhasAux = [];
34   refLinhas = firebase.database().ref('linhas/');
35   searchLinha = '';
36
37   constructor(nav: Nav, private auth: AuthService) {
38     this.nav = nav;
39     this.ref.orderByChild('uid').equalTo(this.auth.getUID()).on('value', resp => {
40       this.favoritos = [];
41       this.favoritos = snapshotToArray(resp);
42     });
43     this.refLinhas.once('value', resp => {
44       this.linhas = [];
45       this.linhasAux = [];
46       this.linhas = snapshotToArray(resp);
47       this.linhasAux = snapshotToArray(resp);
48     });
49   }
50

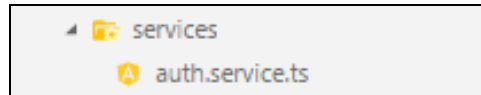
```

Fonte: elaborado pelo autor.

Como serviço de autenticação de usuários para esta aplicação, foi escolhido o serviço Firebase Authentication. Esta ferramenta permite que sejam criados, armazenados e autenticados na aplicação usuários simples, sem necessidade de criação de uma estrutura específica para usuários. No caso da aplicação desenvolvida, não é necessário o armazenamento de muitos dados do usuário, e por este motivo foi possível a utilização do serviço de autenticação do Firebase sem necessidade de criação de uma estrutura complementar dentro da base de dados. Para os usuários deste aplicativo, é possível salvar o e-mail de cadastro, senha e nome de usuário. O Firebase Authentication permite que seja armazenado no próprio serviço, além das propriedades citadas anteriormente, também o endereço para uma foto de perfil caso necessário. Para o gerenciamento das autenticações na aplicação, foi criado um serviço na estrutura do *frontend*, conforme Figura 9. Este serviço realiza as conexões através da API com o Firebase Authentication e manipula as informações

dos usuários conforme necessário, sendo utilizado principalmente nas páginas de autenticação e cadastro.

Figura 9 - Serviço de autenticação



Fonte: elaborado pelo autor.

Assim como na utilização da API para o banco de dados, é necessário primeiro realizar a importação do módulo Firebase para utilização das funções que realizam o intermédio com o serviço da Google. O Quadro 9 ilustra parte do código-fonte do serviço interno de autenticação, demonstrando os métodos `signInWithEmail` (linhas 16 a 19), responsável pela conexão do usuário quando o mesmo já possui cadastro e insere suas informações de login para acessar o aplicativo, e `signUp` (linhas 21 a 23), responsável pela criação de novos usuários quando os mesmos realizarem o cadastro na aplicação.

Quadro 9 - Métodos de autenticação

```

3 import * as firebase from 'firebase/app';
4 import AuthProvider = firebase.auth.AuthProvider;
5
6 @Injectable()
7 export class AuthService {
8   private user: firebase.User;
9
10  constructor(public afAuth: AngularFireAuth) {
11    afAuth.authState.subscribe(user => {
12      | this.user = user;
13    });
14  }
15
16  signInWithEmail(credentials) {
17    console.log('Sign in with email');
18    return this.afAuth.auth.signInWithEmailAndPassword(credentials.email, credentials.password);
19  }
20
21  signUp(credentials) {
22    return this.afAuth.auth.createUserWithEmailAndPassword(credentials.email, credentials.password);
23  }

```

Fonte: elaborado pelo autor.

O método `signInWithEmail` utiliza a função `signInWithEmailAndPassword` (linha 18) da API do Firebase, passando como parâmetros o e-mail e senha informados pelo usuário no respectivo formulário de login. O método `signUp` utiliza a função `createUserWithEmailAndPassword` da API, passando como parâmetros também o e-mail e senha informados pelo usuário no formulário de cadastro. A associação do nome de usuário ao usuário criado é realizada através de outro método no script de controle próprio da tela de cadastro (Quadro 10). Após a chamada do método que realiza a inserção do usuário no Firebase Authentication (linha 40), é realizada uma chamada para o método da API que atualiza o perfil do mesmo usuário (linha 42), inserindo um nome de usuário (linha 43) e,

neste caso, um endereço de foto nulo (linha 44), já que a aplicação não utiliza esta informação.

Quadro 10 - Script da tela de cadastro

```

34  signup() {
35      let data = this.form.value;
36      let credentials = {
37          email: data.email,
38          password: data.password
39      };
40      this.auth.signUp(credentials).then(
41          () => {
42              firebase.auth().currentUser.updateProfile({
43                  displayName: data.username,
44                  photoURL: null
45              }).then(function() {
46                  // Update successful.
47              }).catch(function(error) {
48                  console.log(error);
49              });
50              this.navCtrl.setRoot(HomePage);
51          },
52          error => (this.signupError = error.message)
53      );
54  }
55  }

```

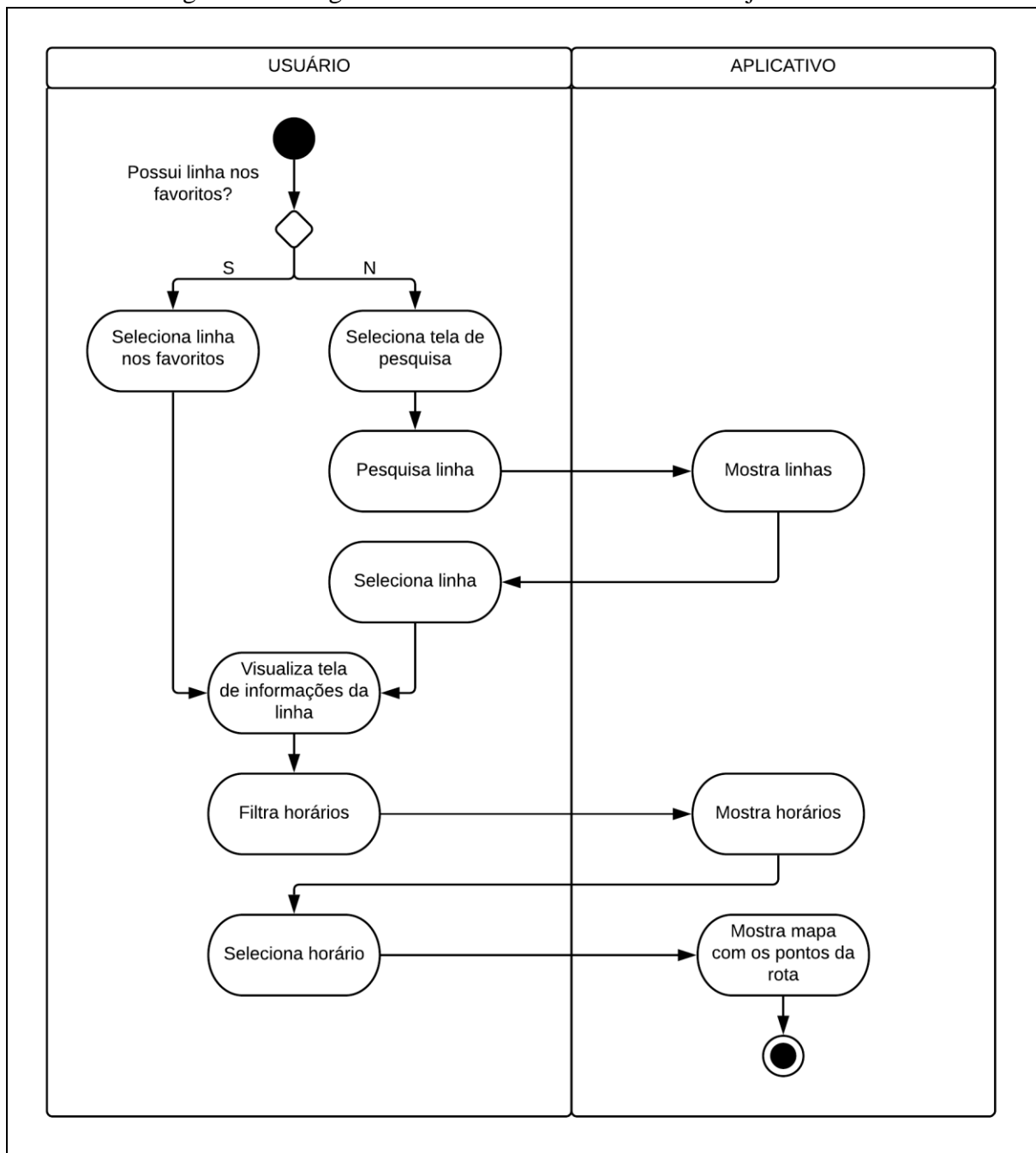
Fonte: elaborado pelo autor.

3.3.3 Operacionalidade da implementação

Nesta seção é apresentada a operacionalidade da implementação do aplicativo BlumenBus, demonstrando o fluxo apresentado no diagrama de atividades. A Figura 10 ilustra o diagrama de atividades referente à consulta do trajeto de uma linha de ônibus, contendo o único ator do sistema: Usuário. Este fluxo se refere ao requisito funcional RF03 e ao caso de uso UC06.

No início do diagrama apresentado, o Usuário decide se consultará uma linha que possui em sua lista de favoritos ou outra linha a ser pesquisada. Caso o Usuário opte por consultar uma linha favorita, ele selecionará a mesma e irá para a tela de informações da linha de ônibus. Caso deseje pesquisar por uma linha, o Usuário deve selecionar a tela de pesquisa. Em seguida, pesquisa a linha desejada. O Aplicativo mostra ao Usuário as linhas que atendam à pesquisa realizada pelo mesmo. O Usuário, em seguida, seleciona a linha escolhida da lista, e após isto o Usuário visualiza a tela de informações da linha de ônibus. O Usuário filtra os horários. Em seguida o Aplicativo mostra os horários. O Usuário seleciona um horário. O Aplicativo então mostra o mapa com os pontos da rota.

Figura 10 - Diagrama de atividades da consulta ao trajeto da linha



Fonte: elaborado pelo autor.

As telas principais do aplicativo são ilustradas na Figura 11 para exemplificar melhor o fluxo demonstrado no diagrama. A partir da tela (a), o usuário poderá decidir se deseja consultar as informações referentes à uma linha de sua lista de favoritos, ou se será necessário pesquisar por outra linha que não esteja em seus favoritos. Esta tela possui um botão que permite ao usuário alternar entre as telas de favoritos (Figura 11 (a)) e de linhas (Figura 11 (b)). A tela de linhas, por sua vez, possui um campo de pesquisa em que o usuário pode realizar a pesquisa através do número ou descrição da linha de ônibus.

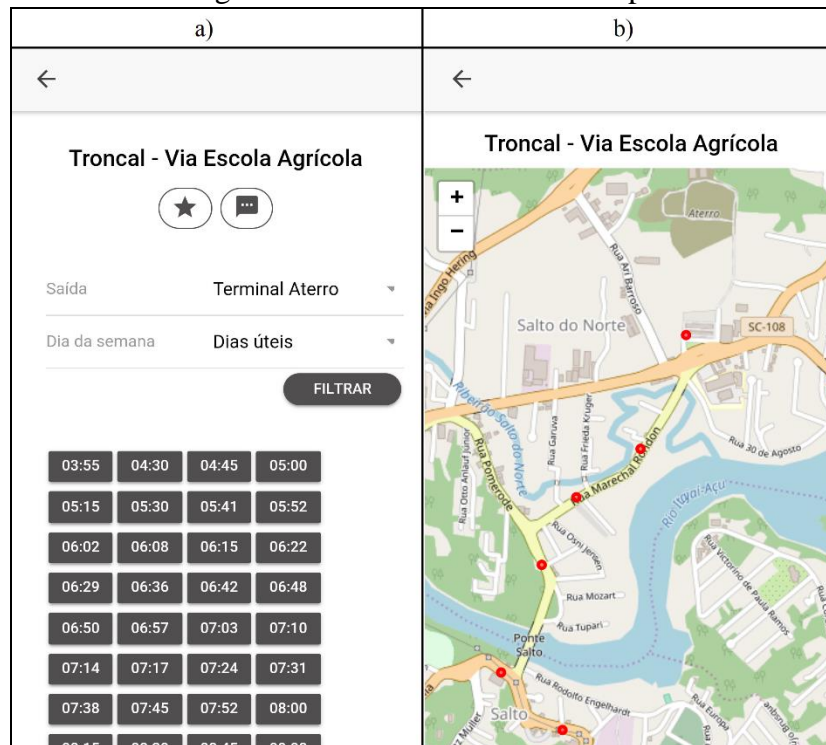
Figura 11 - Telas principais do aplicativo



Fonte: elaborado pelo autor.

Após isto, deve ser selecionada a linha desejada através do toque sobre o respectivo item na lista apresentada. A aplicação então redireciona o usuário para a tela de informações da linha de ônibus, a qual o mesmo selecionou (Figura 12 (a)). Nesta tela, é necessário selecionar um ponto de saída e um dia da semana (dias úteis, sábados e domingos e feriados) e então tocar sobre o botão *Filtrar* para que o sistema apresente em tela a lista contendo os horários de partida da linha de acordo com o filtro utilizado pelo usuário. Caso não existam horários cadastrados na base de dados para o filtro selecionado, será apresentada em tela uma mensagem indicativa. Na listagem de horários, é necessário que o usuário escolha o horário de sua preferência ou necessidade e toque sobre o mesmo para que a aplicação redirecione para a tela do mapa (Figura 12 (b)). Na tela do mapa, o usuário poderá consultar o trajeto que a linha de ônibus realiza naquele horário. Nesta tela, são indicados e marcados através de círculos todos os pontos de ônibus pelos quais a linha realiza paradas até chegar ao seu destino. O usuário pode navegar no mapa, ao deslizar o toque sobre a tela, para seguir com a visualização sobre o trajeto da linha de ônibus ou outros pontos no mapa de seu interesse. Também é possível ao usuário que dê zoom no mapa através do gesto de pinça com dos dedos sobre o mapa.

Figura 12 - Telas de horários e mapa



Fonte: elaborado pelo autor.

3.4 RESULTADOS E DISCUSSÕES

Nesta seção é realizada uma comparação com os trabalhos correlatos, discutida e detalhada a avaliação heurística realizada sobre a aplicação desenvolvida e os resultados provenientes desta avaliação. Desta forma, na seção 3.4.1 é realizada a comparação entre os trabalhos correlatos e o trabalho final desenvolvido. Na seção 3.4.2 é descrito o processo da aplicação da avaliação de usabilidade através de heurísticas. Na seção 3.4.3 é realizada a análise dos resultados obtidos através da avaliação.

3.4.1 Comparação com os trabalhos correlatos

No Quadro 11 demonstra-se uma comparação entre os trabalhos correlatos apresentados na seção 2.4 e o trabalho desenvolvido, destacando suas principais características. É destacado o fato de que todos os trabalhos correlatos e a aplicação desenvolvida neste trabalho possuem como objetivo a disponibilização de linhas e horários de ônibus para consulta através de uma aplicação móvel. Também é possível observar que os 3 trabalhos citados visam oferecer ao usuário uma experiência baseada na rapidez das consultas realizadas com foco na simplicidade em sua interface, assim como o aplicativo BlumenBus.

Quadro 11 - Comparativo trabalhos correlatos

Características/Trabalhos relacionados	BluBus (JENSEN, 2017)	MovelBus (MOVELBUS, 2017)	BusMaps (MOREIRA, 2013)	BlumenBus (KOEPEL, 2018)
Permite a consulta baseada em pesquisa	Sim	Sim	Sim	Sim
Permite a visualização de ressalvas referentes ao trajeto	Sim	Sim	Não	Não
Permite a visualização das rotas através de mapas	Não	Não	Sim	Sim
Fornecer meios para troca de informação entre usuários	Não	Não	Não	Sim
Fornecer alertas de acordo com o horário selecionado	Sim	Não	Não	Não
Permite salvar linha como favorita	Sim	Sim	Sim	Sim
Plataformas	Android	Android	Android e Wap	Android

Fonte: elaborado pelo autor.

Em uma primeira análise sobre o Quadro 11, pode-se notar que o aplicativo BluBus e o aplicativo desenvolvido se destacam em relação a quantidade de recursos oferecidos para o usuário. Nota-se que o maior diferencial do BluBus é o fornecimento de notificações como alerta para o usuário a respeito de uma linha pré-selecionada, já do BlumenBus é o fornecimento de meios para troca de informações entre os usuários. Além da possibilidade de consulta das informações do transporte a partir de uma pesquisa, todos os três trabalhos relacionados e o trabalho desenvolvido se importaram com a disponibilização para o usuário de uma forma de salvar linhas como favoritas, a fim de facilitar a busca e consulta de dados sobre as mesmas. Ambos os aplicativos BusMaps e BlumenBus buscam oferecer ao usuário uma forma de consulta do trajeto realizado pelas linhas de ônibus, enquanto os aplicativos BluBus e MovelBus focam apenas no seu objetivo básico de fornecimento dos horários das linhas, sem se aventurar na utilização de mapas. Além de fornecer as informações necessárias para os usuários de transporte coletivo, os aplicativos BusMaps e BlumenBus oferecem a visualização das rotas das linhas de ônibus a fim de prover um maior conhecimento sobre os trajetos oferecidos pelas linhas disponíveis. Apenas o aplicativo desenvolvido neste trabalho fornece uma interface em que seja possível a colaboração entre usuários, conforme visto no Quadro 11.

Nenhum dos trabalhos relacionados apresenta a utilização de uma interface em que seja possível a colaboração entre os usuários. Tendo como base o cenário demonstrado, o trabalho desenvolvido oferece o mesmo foco sobre as informações do transporte coletivo que os demais trabalhos relacionados apresentados, que possuem relevância para grande parte da

população de Blumenau/SC. A partir da utilização de recursos comuns com as ferramentas apresentadas como trabalhos correlatos, foi adicionada em seu funcionamento a essência da colaboratividade. Sendo assim, o trabalho realizado tem como finalidade, além da consulta sobre horários e trajetos, que os usuários possam contribuir com informações em tempo real sobre as linhas que estão utilizando. Desta forma, o aplicativo desenvolvido visa não apenas a disponibilização de dados fixos, como também a aprimoração da base de conhecimento através da adição espontânea de informações dinâmicas e específicas do momento relatado.

Além do descrito, é possível notar que dentre todos os trabalhos apresentados, apenas o BlumenBus garante a publicação para a plataforma iOS, utilizando a mesma implementação. Isso porque o aplicativo é desenvolvido utilizando o Ionic e a plataforma Cordova que permite a geração de um aplicativo para iOS.

3.4.2 Avaliação heurística

A avaliação heurística é um método de inspeção de usabilidade em que a avaliação é realizada por especialistas com base em um conjunto de heurísticas, em busca de problemas que prejudiquem a usabilidade (NIELSEN; MACK, 1994 apud MANTAU et al; 2013, p. 1). Dessa forma, para avaliar a usabilidade do aplicativo desenvolvido realizou-se uma avaliação heurística com dois usuários. Esses usuários foram considerados especialistas por entenderem de aplicativos em dispositivos Android e compreenderem as regras de negócio relacionadas ao transporte público. Ainda, um dos especialistas era um acadêmico do curso de Ciência da Computação que já realizou disciplinas na área de desenvolvimento para aplicativos móveis.

O aplicativo desenvolvido foi disponibilizado para os avaliadores através de uma URL pela qual foi possível realizar o download da aplicação, instalação em dispositivo móvel e então a utilização do aplicativo para avaliação. A avaliação heurística foi realizada através de um formulário elaborado pelo autor na ferramenta de formulários do Google, que pode ser visualizado no Apêndice A. A seguir são descritas as heurísticas utilizadas no processo de avaliação, adaptadas de Nielsen e Mack (1994) por Mantau et al. (2013), que tem como objetivo avaliar sistemas móveis com funcionalidades colaborativas.

- a) H1 - Visibilidade do estado do sistema: o aplicativo deve fornecer feedback adequado aos usuários dentro de um tempo razoável;
- b) H2 - Compatibilidade do sistema com o mundo real: o sistema deve usar termos familiares ao usuário ao invés de termos orientados ao software;
- c) H3 - Consistência e mapeamento: o modelo conceitual de interação do usuário com o aplicativo deve estar de acordo com o contexto;

- d) H4 - Reconhecimento ao invés de memorização: o aplicativo deve conter instruções visíveis ou de fácil recuperação quando necessárias;
- e) H5 - Flexibilidade e eficiência de uso: o aplicativo deve prover meios para usuários experientes acelerarem a interação e de apoiar usuários novatos;
- f) H6 – Design estético e minimalista: o aplicativo deve exibir apenas as informações que sejam importantes e necessárias;
- g) H7 - Gerenciamento de erros: o aplicativo deve emitir mensagens de erro com linguagem clara indicando o problema e construtivamente sugerindo uma solução;
- h) H8 - Facilidade de entrada, visualização, e leitura da tela: os dados apresentados na tela devem ser fáceis de serem lidos e navegados;
- i) H9 - Convenções estéticas, sociais e privadas: Levar em consideração aspectos estéticos e emocionais a serem apresentados nos dispositivos móveis. Deixar claro que as informações do usuário estão seguras;
- j) H10 - Fornecer comunicação de artefatos compartilhados: o aplicativo deve disponibilizar informações sobre as ações dos outros usuários;
- k) H11 - Fornecer proteção: o aplicativo deve conter mecanismos para evitar conflitos ao acesso simultâneo a um mesmo conjunto de artefatos;
- l) H12 - Gerenciamento de colaboração de baixo e alto acoplamento: o acoplamento é o grau que as pessoas trabalham em conjunto, e representa a quantidade de trabalho que uma pessoa pode realizar antes de precisar discutir, consultar ou pedir uma informação para outra pessoa;
- m) H13 - Permitir que as pessoas coordenem suas ações: o aplicativo deve permitir que as tarefas sejam realizadas na ordem certa, no momento certo, sem ignorar as restrições impostas.

O processo de avaliação foi realizado em duas etapas. Na primeira etapa cada especialista passou por pelo menos 30 minutos inspecionando o aplicativo. Na segunda etapa os avaliadores responderam as 13 heurísticas, descritas anteriormente, informando se o aplicativo BlumenBus atendia, atendia parcialmente ou não atendia a cada heurística, e caso avaliassem que a aplicação atendia parcialmente ou não atendia à alguma heurística, informavam a gravidade do problema encontrado e justificavam sua resposta. A gravidade varia de 0 a 4, sendo: 0) não é encarado como um problema; 1) problema estético que não tem necessidade de ser corrigido; 2) problema pequeno com baixa prioridade na correção; 3) problema grande, com alta prioridade de correção; 4) problema catastrófico, onde é

indispensável sua correção (NIELSEN; MACK, 1994 apud MANTAU et al; 2013, p. 6). Durante as respostas, os avaliadores tinham a possibilidade de voltar a usar o aplicativo caso julgassem necessário.

3.4.3 Resultados

A avaliação foi realizada com o propósito de identificar problemas de usabilidade do sistema e o seu grau de gravidade. O Quadro 12 apresenta o resumo das heurísticas, problemas e gravidade encontrados na avaliação.

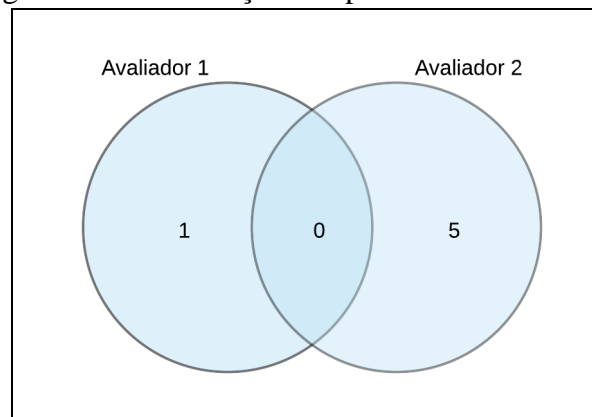
Quadro 12 - Resumo das heurísticas, problemas e gravidade

Heurística	Problema	Gravidade
H1	1	2
H2	--	--
H3	--	--
H4	2	1
H5	--	--
H6	--	--
H7	3 e 4	2
H8	5	1
H9	6	4
H10	--	--
H11	--	--
H12	--	--
H13	--	--

Fonte: elaborado pelo autor.

A coluna Problemas enumera os problemas que foram encontrados pelos especialistas no decorrer da avaliação. Conforme indica o Quadro 12, para as heurísticas H2, H3, H5, H6, H10, H11, H12 e H13 não foi encontrado nenhum problema de usabilidade, demonstrando que o aplicativo desenvolvido está de acordo com o proposto por estas heurísticas. Contudo, para as heurísticas H1, H4, H7, H8 e H9, foram encontrados problemas pelos avaliadores, sendo que um destes problemas foi classificado como gravidade 4. No total, os avaliadores encontraram 6 problemas de usabilidade no BlumenBus. O número dos problemas identificado na coluna central (Problemas) corresponde a descrição dos problemas apresentados no decorrer desses resultados, no Quadro 13. A Figura 13 apresenta a distribuição dos problemas encontrados para cada um dos avaliadores.

Figura 13 - Distribuição dos problemas encontrados



Fonte: elaborado pelo autor.

Como demonstrado na Figura 13, o primeiro avaliador encontrou apenas um problema (Problema 3) e o segundo avaliador encontrou 5 problemas (Problemas 1, 2, 4, 5 e 6). Ambos os avaliadores encontraram problema referente a heurística H7, sendo que o Avaliador 2 também julgou necessárias correções referentes às heurísticas H1, H4, H8 e H9. Os principais problemas encontrados estão listados no Quadro 13, bem como a avaliação se o aplicativo atende, atende parcialmente ou não atende à heurística, e a justificativa do avaliador.

Quadro 13 - Problemas e justificativas

Problema 1	Atende parcialmente	O mapa apresentou problemas de visualização por conta da rede do celular e não informou a causa deste problema, mas a qualidade da rede é algo fora do controle do desenvolvedor ao meu ver.
Problema 2	Atende parcialmente	Inicialmente não existe uma orientação, porém a disposição de tela e ícone dos botões são autoexplicativos.
Problema 3	Não atende	Seria necessário aparecer uma mensagem em caso de mal conexão com a internet.
Problema 4	Atende parcialmente	Houve uma falha na visualização do mapa por conta de má conexão, porém outros campos de preenchimento estão bem tratados como por exemplo a seleção da rota da linha.
Problema 5	Atende parcialmente	As informações podiam conter uma fonte só um pouco maior do atual.
Problema 6	Não atende	Por ser um aplicativo de mapas e conter cadastro, poderia conter uma explicação inicial sobre qual a procedência dos dados do usuário.

Fonte: elaborado pelo autor.

4 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um aplicativo para dispositivos móveis para consulta de informações referentes ao transporte público de Blumenau/SC. O desenvolvimento da aplicação foi realizado através do editor Visual Studio Code, sob a plataforma Cordova junto à biblioteca Ionic, que possibilitou a implementação sem necessidade de programação direta na linguagem Android. Para o *frontend* também foi utilizado o *framework* Angular, a biblioteca Leaflet e o interpretador Node.js. Para o *backend* foi utilizado o serviço Firebase da Google, para banco de dados e autenticação de usuários.

O trabalho teve como objetivo principal desenvolver um aplicativo móvel para disponibilizar informações do transporte público. Pode-se afirmar que o objetivo foi alcançado, pois ao utilizar o aplicativo desenvolvido é possível consultar informações referentes às linhas de ônibus do transporte público de Blumenau/SC, previamente cadastradas na base de dados.

O objetivo específico de fornecer uma interface para colaboração entre os usuários do aplicativo foi atendido. Houve a preocupação no desenvolvimento de criar uma tela de colaboração em que os usuários podem realizar comentários sobre determinada linha de ônibus e consultar comentários realizados por outros usuários da aplicação. A comparação com os trabalhos correlatos também demonstra que foi implementado um diferencial no trabalho desenvolvido, que permite a comunicação entre usuários.

Tendo em vista a avaliação heurística realizada com dois avaliadores para testar a usabilidade do aplicativo, pode-se concluir que o objetivo específico de disponibilizar interfaces gráficas mantendo requisitos e padrões de usabilidade aplicados a dispositivos móveis com base nas diretrizes do Android e do Material Design foi atendido. A avaliação heurística mostrou bons resultados, porém também apontou a necessidade de melhoria significativa em alguns pontos da interação com o usuário. Contudo, os pontos destacados relacionam-se em sua maioria sobre a conectividade, sendo um fator da usabilidade não atrelado à interface gráfica, porém que deve ser melhorado.

Durante o processo de desenvolvimento do aplicativo houve uma grande carga de conhecimento adquirida, considerando que este trabalho foi o primeiro aplicativo construído do zero pelo autor. Foi necessária a realização de inúmeras pesquisas e leituras sobre as ferramentas e tecnologias utilizadas durante a implementação. Ferramentas como o Firebase da Google auxiliaram neste sentido, tendo em vista a falta de alguns conhecimentos relacionados à tecnologias de *backend*. Houve momentos em que foi necessário trocar uma

tecnologia que havia sido definida no início do trabalho por outra, após considerar as limitações que seriam impostas ao aplicativo. Este obstáculo acabou causando uma série de atrasos nos prazos que inicialmente haviam sido definidos para auxiliar no processo de implementação.

Conforme observado, o trabalho desenvolvido possui limitações como alguns problemas relacionados à interação do aplicativo com o usuário em momentos que há falta de conectividade com a internet. Como se trata de um aplicativo desenvolvido sob um *backend* totalmente online, se faz necessário manter a conexão ativa para que as informações disponibilizadas possam ser consultadas pelo usuário, porém quando não é possível realizar esta conexão com a internet, se torna interessante que o usuário seja informado desta situação.

É possível concluir que este trabalho pode vir a contribuir socialmente para que os usuários do transporte público de Blumenau/SC tenham uma maneira mais eficiente e rápida de realizar consultas às informações das linhas de ônibus, além de possibilitar a disseminação de informações através da colaboração entre usuários, auxiliando na rotina das pessoas que dependem deste meio de transporte. Além de seu lado social, o trabalho apresenta contribuições tecnológicas no que diz respeito a apresentação do trajeto das linhas de ônibus através de um mapa, utilizando o OpenStreetMap, que por si também é considerado uma plataforma colaborativa. Desta forma, o usuário pode obter conhecimento do caminho utilizado por linhas do transporte público que não está acostumado a utilizar em seu dia-a-dia.

4.1 EXTENSÕES

Como sugestão de extensão para o trabalho, propõe-se:

- a) aprimoramento na interação do sistema para com o usuário, de forma a melhorar a apresentação de informações que controlem possíveis erros na aplicação, como no caso de falta de internet ou atraso da mesma;
- b) sistema de notificações para o usuário, conforme linha(s) previamente marcada(s), para que seja informado quando houver novos comentários ou atualização de informações da linha de ônibus;
- c) aprimoramento do sistema de apresentação de trajeto em mapa, para que o sistema possa lidar com situações como: alteração no trajeto em determinados horários e alteração nos pontos de parada em horários específicos.
- d) aprimoramento do mecanismo de comentários, de forma que seja possível replicar comentários já existentes, bem como a criação de uma rotina que apague comentários muito antigos (após um período definido), e melhora na visualização

dos comentários aplicando paginação nesta tela.

- e) implementação de um modo offline, que possibilite o usuário consultar as informações mesmo que não esteja conectado à internet, porém informando-o que estas informações podem estar desatualizadas devido à falta de conectividade.
- f) implementação de um módulo administrador, para gerenciamento das informações disponibilizadas no aplicativo.

REFERÊNCIAS

- CAMARGO, Aline. **Blumob tem avaliação positiva após um mês de serviço em Blumenau**. Jornal de Santa Catarina. [s.i], p. 1-1. 07 ago. 2017. Disponível em: <<http://jornaldesantacatarina.clicrbs.com.br/sc/geral/noticia/2017/08/blumob-tem-avaliacao-positiva-apos-um-mes-de-servico-em-blumenau-9863434.html>>. Acesso em: 14 set. 2017.
- CASTRO, Alberto; MENEZES, Crediné. Aprendizagem colaborativa com suporte computacional. In: PIMENTEL, Mariano; FUKS, Hugo (Org.). **Sistemas Colaborativos**. Rio de Janeiro: Elsevier, 2012. Cap. 9. p. 135-153.
- CETIC. **J2 – Indivíduos que possuem telefone celular**. [S.I.], 2016. Disponível em: <<http://www.cetic.br/tics/usuarios/2016/total-brasil/J2/>>. Acesso em: 15 set. 2017.
- ELLIS, C. A.; GIBBS, S. J.; REIN, G. Groupware: some issues and experiences. *Communications of the ACM*, v. 34, n. 1, p. 39-58, jan. 1991.
- FAGUNDES, L.C., SATO, L.S., MAÇADA, D. L. **Aprendizes do Futuro – as inovações já começaram**. Brasília; MEC, 1999.
- FERRIS, B. **OneBusAway: Improving the Usability of Public Transit**. 2011. 234 f.. Tese (Doutorado) - Department of Computer Science and Engineering, University of Washington, Washington, 2011. Disponível em: <<https://www.cs.washington.edu/sites/default/files/hci/papers/ferrisdissertation.pdf>>. Acesso em: 11 set. 2017.
- FUKS, Hugo et al. Teorias e modelos de colaboração. In: PIMENTEL, Mariano; FUKS, Hugo (Org.). **Sistemas Colaborativos**. Rio de Janeiro: Elsevier, 2012. Cap. 2. p. 16-33.
- JENSEN, Claus. **Estudantes criam o BluBus, aplicativo para consulta de horários de ônibus em Blumenau**; O Blumenauense, 2017. Disponível em: <<http://www.oblumenauense.com.br/site/estudantes-criam-o-blubus-aplicativo-para-consultade-horarios-de-onibus-em-blumenau/>>. Acesso em: 11 set. 2017.
- MANTAU, Márcio J. et al. **Avaliação heurística para groupwares móveis: um estudo de caso utilizando um audience response system**. Joinville, 2013.
- MOREIRA, Matheus Villela. **Aplicativo para plataforma Android: "BusMaps"**, informações das linhas de ônibus da cidade de Florianópolis. 2013. 43 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 2013. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_1567/TCC2-13-12-2013.pdf>. Acesso em: 05 set. 2017.
- MOVELBUS. **MovelBus**. [S.I.], 2017. Disponível em: <<http://www.movelbus.com.br/inicio>>. Acesso em: 12 set. 2017.
- NICOLACI-DA-COSTA, Ana Maria; PIMENTEL, Mariano. Sistemas colaborativos para uma nova sociedade e um novo ser humano. In: PIMENTEL, Mariano; FUKS, Hugo (Org.). **Sistemas Colaborativos**. Rio de Janeiro: Elsevier, 2012. Cap. 1. p. 3-15.
- NIELSEN, Jakob; MACK, Robert L. **Usability Inspection Methods**. New York: John Wiley & Sons, 1994. 188 p.
- PENA, Rodolfo F. Alves. **Problemas no transporte público**; [S.I.], Brasil Escola, [2013?]. Disponível em <<http://brasilecola.uol.com.br/geografia/problemas-no-transporte-publico.htm>>. Acesso em 15 set. 2017.

PIMENTEL, Mariano; GEROSA, Marco Aurélio; FUKS, Hugo. Sistemas de comunicação para colaboração. In: PIMENTEL, Mariano; FUKS, Hugo (Org.). **Sistemas Colaborativos**. Rio de Janeiro: Elsevier, 2012. Cap. 5. p. 65-93.

SETERB. **Transporte Coletivo**. Blumenau, [2017?]. Disponível em:
<<http://www.blumenau.sc.gov.br/governo/seterb/pagina/transportes-seterb//coletivo-seterb>>. Acesso em: 11 set. 2017.

VIVACQUA, Adriana Santarosa; GARCIA, Ana Cristina Bicharra. Ontologia de colaboração. In: PIMENTEL, Mariano; FUKS, Hugo (Org.). **Sistemas Colaborativos**. Rio de Janeiro: Elsevier, 2012. Cap. 3. p. 34-49.

APÊNDICE A – Formulário de avaliação

Este apêndice contém o formulário de avaliação heurística apresentado aos especialistas. Este conteúdo foi gerado com o auxílio da ferramenta Google Forms.

Figura 14 - Introdução ao questionário

Avaliação de usabilidade BlumenBus

O aplicativo móvel avaliado é um trabalho de conclusão do curso de Bacharelado em Sistemas de Informação, na instituição de ensino Universidade Regional de Blumenau (FURB), no 2º semestre de 2018.

Este aplicativo não está disponível publicamente, sendo acessado através de download do arquivo de aplicação (.apk) em url disponibilizada pelo autor e instalação em smartphone dos participantes.

Na próxima seção, serão apresentados os objetivos da aplicação avaliada.

PRÓXIMA

Nunca envie senhas pelo Formulários Google.

Fonte: elaborado pelo autor.

Figura 15 - Objetivos da aplicação do questionário

Avaliação de usabilidade BlumenBus

Objetivos da aplicação

Esta avaliação tem por objetivo avaliar o aplicativo BlumenBus, avaliando sua usabilidade e propondo soluções para possíveis problemas encontrados.

Esta avaliação servirá como base para as futuras melhorias e mudanças pelas quais a aplicação avaliada possa vir a passar, e também levantar a viabilidade da continuação do projeto.

VOLTAR **PRÓXIMA**

Nunca envie senhas pelo Formulários Google.

Fonte: elaborado pelo autor.

Figura 16 - Termo de consentimento do questionário

Termo de consentimento

Eu, usuário que está avaliando este projeto, estou sendo convidado a participar de um estudo denominado Avaliação de usabilidade do aplicativo BlumenBus, cujos objetivos e justificativas são: avaliar a aplicação mencionada a partir da utilização do aplicativo e, posteriormente, da realização da avaliação de usabilidade. Esta avaliação servirá como base para as futuras melhorias e mudanças pelas quais a aplicação avaliada possa vir a passar, e também levantar a viabilidade da continuação do projeto.

A minha participação no referido estudo será no sentido de executar o aplicativo BlumenBus, utilizar o aplicativo por no mínimo 30 minutos e executar a avaliação heurística da aplicação por meio de um formulário de perguntas definidas.

Recebi os esclarecimentos necessários sobre os possíveis desconfortos e riscos decorrentes do estudo, levando-se em conta que é uma pesquisa, e os resultados positivos ou negativos somente serão obtidos após a sua realização. Assim, estou sujeito a utilização do aplicativo Andaê por no mínimo 30 minutos. Ainda, a minha avaliação poderá ou não ser considerada no resultado final da aplicação, dependendo de como eu irei responder à avaliação.

Estou ciente de que a minha privacidade será respeitada, ou seja, meu nome ou qualquer outro dado ou elemento que possa, de qualquer forma, me identificar, será mantido em sigilo. Também fui informado de que posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar, e que por desejar sair da pesquisa, não sofrerei qualquer prejuízo.

Os pesquisadores envolvidos com o referido projeto são: Marcela Zanatta Koepsel, da Universidade Regional de Blumenau (FURB), com a qual posso entrar em contato pelo e-mail marcelaaazk@hotmail.com e Professora Luciana Pereira de Araújo Kohler, da Universidade Regional de Blumenau (FURB), com a qual posso entrar em contato pelo e-mail jpa@furb.br.

Fonte: elaborado pelo autor.

Figura 17 - Parte dois do termo de consentimento

É assegurada a assistência antes e depois da avaliação, bem como me é garantido o livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências.

Enfim, tendo sido orientado quanto ao teor de todo o aqui mencionado, e compreendido a natureza e o objetivo do referido estudo, manifesto me livre consentimento em participar, estando totalmente ciente de que não há nenhum valor econômico, a receber ou pagar, por minha participação.

Em caso de reclamação ou qualquer tipo de denúncia sobre este estudo devo entrar em contato com Professora Luciana Pereira de Araújo Kohler, da Universidade Regional de Blumenau (FURB), através do e-mail jpa@furb.br.

Blumenau, 29 de novembro de 2018
Marcela Zanatta Koepsel, Acadêmico - Universidade Regional de Blumenau (FURB)

Você concorda com termo de consentimento? *

Sim

Não

Fonte: elaborado pelo autor.

Figura 18 - Avaliação Heurística

Avaliação Heurística

É uma técnica de inspeção de usabilidade utilizando um conjunto de heurísticas em que especialistas avaliam se os elementos da interface com o usuário estão de acordo com os princípios.

Tem como objetivo antever potenciais problemas, buscar por problemas de usabilidade e propor soluções para elas.

A avaliação é composta por três etapas:

Avaliação
Cada especialista passa por no mínimo 30 minutos inspecionando o aplicativo. O avaliador deve olhar todos os detalhes de forma minuciosa, identificando problemas potenciais de usabilidade. É interessante que o avaliador utilize cada tela mais de uma vez para que possa sentir o fluxo de interação e o escopo do aplicativo, e também poder focar em elementos específicos da interface.

Responder questionário
Os avaliadores respondem um conjunto de 13 heurísticas que se encontram nas seções a seguir, informando se o aplicativo Andaê atende, atende parcialmente ou não atende a elas. Para cada heurística atendida parcialmente ou não atendida, os avaliadores informam uma a gravidade do problema encontrado e uma justificativa. A gravidade varia de 0 a 4, sendo:

- 0) Não é encarado como um problema;
- 1) Problema estético que não tem necessidade de ser corrigido;
- 2) Problema pequeno com baixa prioridade na correção;
- 3) Problema grande, com alta prioridade de correção;
- 4) Problema catastrófico, onde é indispensável sua correção.

O avaliador pode voltar a testar o aplicativo caso necessite e é imprescindível que a avaliação seja feita de forma sincera para que possa ser validada.

Discussão dos resultados
Os avaliadores se reúnem a fim de discutir o que descobriram, priorizam os problemas que encontraram e sugerem soluções.

Fonte: elaborado pelo autor.

Figura 19 - Heurística H1

H1 - Visibilidade do estado do sistema

O sistema deve fornecer feedback adequado aos usuários dentro de um tempo razoável. Através do dispositivo móvel o sistema deve manter o usuário informado sobre o que está ocorrendo. Além disso, o sistema deve priorizar mensagens a respeito de informações críticas e de contexto, como status da rede, bateria e condições do ambiente (Nielsen e Mack, 1994; Bertini et al., 2009).

O aplicativo Andaê atende a heurística H1? *

Atende
 Atende Parcialmente
 Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 20 - Heurística H2

H2 - Compatibilidade do sistema com o mundo real

O sistema deve usar termos familiares ao usuário ao invés de termos orientados ao software. Devem ser seguidas convenções do mundo real de modo que as informações apareçam numa ordem sequencial e lógica (Nielsen e Mack, 1994; Bertini et al., 2009). As informações devem ser apresentadas de forma natural e em uma ordem lógica, e sempre que possível o sistema deve ter a capacidade de perceber as mudanças no ambiente e adaptar-se a ele (sistemas conhecidos como contextaware systems).

O aplicativo Andaê atende a heurística H2? *

Atende
 Atende Parcialmente
 Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 21 - Heurística H3

H3 - Consistência e mapeamento

Um usuário não deve adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. O modelo conceitual de interação do usuário com o sistema deve estar de acordo com o contexto. O mapeamento entre as ações e interações do usuário (e.g. controles de navegação) e a tarefa real correspondente (e.g. navegação no mundo real) deve ser consistente (Nielsen e Mack, 1994; Bertini et al., 2009).

O aplicativo Andaê atende a heurística H3? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0	1	2	3	4
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 22 - Heurística H4

H4 - Reconhecimento ao invés de memorização

Tomar objetos, ações e opções visíveis. O usuário não deve ter que lembrar uma informação de uma parte para outra do diálogo. Instruções devem estar visíveis ou ser de fácil recuperação quando necessárias (Nielsen e Mack, 1994).

O aplicativo Andaê atende a heurística H4? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0	1	2	3	4
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 23 - Heurística H5

H5 - Flexibilidade e eficiência de uso

Prover meios para usuários experientes acelerarem a interação e de apoiar usuários novatos. Permitir que os usuários personalizem ações frequentes, bem como configurem o sistema de acordo com a necessidade do contexto (Nielsen e Mack, 1994; Bertini et al., 2009).

O aplicativo Andaê atende a heurística H5? *

Atende
 Atende Parcialmente
 Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 24 - Heurística H6

H6 – Design estético e minimalista

Diálogos não devem conter informações irrelevantes ou raramente necessárias. Exibir apenas as informações que sejam importantes e necessárias. Os recursos de tela são propriedades escassas e devem ser utilizadas com sabedoria (Nielsen e Mack, 1994; Bertini et al., 2009).

O aplicativo Andaê atende a heurística H6? *

Atende
 Atende Parcialmente
 Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 25 - Heurística H7

H7 - Gerenciamento de erros

Mensagens de erro devem ser expressas com linguagem clara indicando o problema e construtivamente sugerindo uma solução. Proteger os usuários dos erros que podem ocorrer. Auxiliar o usuário a reconhecer, diagnosticar e se possível recuperar-se do erro ocorrido (Nielsen e Mack, 1994; Bertini et al., 2009).

O aplicativo Andaê atende a heurística H7? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 26 - Heurística H8

H8 - Facilidade de entrada, visualização, e leitura da tela

Dispositivos móveis devem fornecer meios facilitados para entrada de dados, os dados apresentados na tela devem ser fáceis de serem lidos e navegados. É ideal apresentar apenas informações cruciais sobre o sistema (Bertini et al., 2009).

O aplicativo Andaê atende a heurística H8? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 27 - Heurística H9

H9 - Convenções estéticas, sociais e privativas

Levar em consideração aspectos estéticos e emocionais a serem apresentados nos dispositivos móveis. Deixar claro que as informações do usuário estão seguras (Bertini et al., 2009).

O aplicativo Andaê atende a heurística H9? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 28 - Heurística H10

H10 - Fornecer comunicação de artefatos compartilhados (i.e. Feedthrough)

Uma importante necessidade de um groupware é disponibilizar informações sobre as ações dos outros usuários, reconhecer o que os outros estão fazendo com os artefatos compartilhados (Baker et al., 2001).

O aplicativo Andaê atende a heurística H10? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 29 - Heurística H11

H11 - Fornecer proteção

Em workspaces compartilhados o acesso simultâneo a um mesmo conjunto de artefatos pode ocasionar conflitos. Em alguns casos, as ações de um usuário podem vir a interferir nas atividades dos demais, e assim sendo, o sistema deve disponibilizar mecanismos para que estes conflitos não ocorram (Baker et al., 2001).

O aplicativo Andaê atende a heurística H11? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 30 - Heurística H12

H12 - Gerenciamento de colaboração de baixo e alto acoplamento

O acoplamento é o grau que as pessoas trabalham em conjunto, e representa a quantidade de trabalho que uma pessoa pode realizar antes de precisar discutir, consultar ou pedir uma informação para outra pessoa (Baker et al., 2001).

O aplicativo Andaê atende a heurística H12? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.

Figura 31 - Heurística H13

H13 - Permitir que as pessoas coordenem suas ações

Um dos problemas encontrados ao longo da colaboração face a face é como o grupo media suas interações. Ações de coordenação envolvem realizar as tarefas na ordem certa, no momento certo, sem ignorar as restrições impostas (Baker et al., 2001).

O aplicativo Andaê atende a heurística H13? *

Atende

Atende Parcialmente

Não atende

Gravidade *

0 1 2 3 4

Justificativa

Sua resposta _____

Fonte: elaborado pelo autor.