

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MURILO SANTOS BUENO

**ViajeTriFácil: Um Sistema Baseado em  
Software Livre para Consulta de Trajetos de  
Ônibus Aplicado a Cidade de Porto Alegre**

Trabalho de Graduação.

Prof. Dr. Alexandre da Silva Carissimi  
Orientador

Porto Alegre, julho de 2014.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

## **AGRADECIMENTOS**

Agradeço, acima de tudo, a Deus por me dar a capacidade de perceber que sonhos podem ser alcançados e também por colocar em meu caminho pessoas que me ensinaram e ajudaram no decorrer de minha vida e, em especial, na execução deste trabalho.

Ao meu amigo e orientador, Alexandre da Silva Carissimi, por toda dedicação e apoio prestados, ideias compartilhadas e valorização de todo o trabalho desenvolvido. Sem sua orientação este trabalho não seria possível.

A professora Ana Lucia C. Bazzan pela indicação de materiais de pesquisa, conselhos e comentários valiosos que contribuíram para transcrever e entender melhor sobre assuntos associados à mobilidade urbana, em especial, ao transporte público.

Ao Fernando Trebien, pelas ideias e conhecimento compartilhados, os quais foram de imensa importância, pois se tornaram a base para o desenvolvimento deste trabalho.

Ao colega de trabalho Tiago Fonseca, pela indicação de ferramentas e metodologias extremamente úteis para a construção do sistema proposto.

A minha família que sempre esteve ao meu lado em todos os momentos. Em especial a Daiane Fagundes Ribas, que me deu carinho, teve paciência para aguentar minhas ausências e me incentivou a concluir mais esta etapa em minha vida; aos meus pais que sempre estiveram ao meu lado e são a inspiração para a pessoa que sou hoje; e também a minha tia Anna, pelo tempo investido em meus ensinamentos e pelo amor incondicional que sempre dedicou a mim.



# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>7</b>
<b>LISTA DE FIGURAS</b> .....	<b>9</b>
<b>LISTA DE TABELAS</b> .....	<b>11</b>
<b>RESUMO</b> .....	<b>12</b>
<b>ABSTRACT</b> .....	<b>13</b>
<b>1 INTRODUÇÃO</b> .....	<b>14</b>
1.1 Objetivo.....	16
1.2 Organização do trabalho.....	17
<b>2 CONTEXTUALIZAÇÃO</b> .....	<b>18</b>
2.1 Sistemas de Transporte Inteligentes (ITS).....	18
2.2 Sistemas Avançados de Transporte Público (APTS).....	20
2.3 <i>OpenStreetMap</i> (OSM).....	23
2.4 <i>OpenTripPlanner</i> (OTP).....	27
2.5 Sistemas de Mobilidade Urbana Existentes.....	31
2.6 Considerações Finais.....	37
<b>3 ESPECIFICAÇÃO E PROJETO</b> .....	<b>40</b>
3.1 Requisitos.....	40
3.1.1 Requisitos Funcionais.....	41
3.1.2 Requisitos não funcionais.....	43
3.2 Diagrama de casos de uso.....	44
3.3 Prototipação de interface.....	45
3.3.1 Interface web.....	45
3.3.2 Interface para dispositivos móveis.....	52
3.4 Arquitetura do sistema.....	56
3.4.1 Modelo cliente-servidor.....	56
3.4.2 Padrão <i>Model-View-Controller</i> (MVC).....	58
3.4.3 Modelo de comunicação e tipos de dados fornecidos pelas aplicações.....	59
3.5 Considerações finais.....	59
<b>4 IMPLEMENTAÇÃO</b> .....	<b>60</b>
4.1 Arquitetura completa da solução.....	60
4.2 Implementação no lado servidor.....	61
4.3 Implementação no lado cliente.....	64
4.3.1 Sistema web.....	64
4.3.2 Sistema <i>mobile</i> .....	68
4.4 Considerações finais.....	70
<b>5 VALIDAÇÃO</b> .....	<b>71</b>
5.1 Metodologia.....	71
5.2 Testes funcionais básicos.....	71

5.2.1	Cenário I: Um ônibus do ponto “A” ao ponto “B” .....	72
5.2.2	Cenário II: Dois ônibus do ponto “A” ao ponto “B” .....	74
5.2.3	Cenário III: Três ônibus do ponto “A” ao ponto “B” .....	76
<b>5.3</b>	<b>Considerações parciais .....</b>	<b>77</b>
<b>5.4</b>	<b>Pesquisa de opinião.....</b>	<b>78</b>
5.4.1	Perfil do usuário .....	78
5.4.2	Mobilidade.....	79
5.4.3	Software de mobilidade urbana .....	80
5.4.4	Avaliação do ViajeTriFácil .....	81
5.4.5	Opiniões sobre o ViajeTriFácil .....	82
<b>5.5</b>	<b>Considerações finais .....</b>	<b>84</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>86</b>
	<b>REFERÊNCIAS.....</b>	<b>88</b>
	<b>APÊNDICE QUESTIONÁRIO PARA AVALIAR UM SISTEMA DE CONSULTAS DE TRAJETOS DE ÔNIBUS .....</b>	<b>93</b>

## LISTA DE ABREVIATURAS E SIGLAS

AJAX	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
APTS	<i>Advanced Public Transportation Systems</i>
ARTS	<i>Advanced Rural Transportation Systems</i>
ATIS	<i>Advanced Traveler Information Systems</i>
ATMS	<i>Advanced Transportation Management Systems</i>
AVCS	<i>Advanced Vehicle Control Systems</i>
BRT	<i>Bus Rapid Transit</i>
CC BY-CA	<i>Creative Commons Attribution-ShareAlike 2.0</i>
CSS	<i>Cascading Style Sheets</i>
CVO	<i>Commercial Vehicle Operations</i>
DENATRAN	Departamento Nacional de Trânsito
EPTC	Empresa Pública de Transporte e Circulação
EPS	<i>Electronic Payment Systems</i>
FMS	<i>Fleet Management System</i>
GIS	<i>Geographical Information System</i>
GPL	<i>GNU General Public License</i>
GPS	<i>Global Positioning System</i>
GPX	<i>GPS eXchange Format</i>
GTFS	<i>General Transit Feed Specification</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IP	<i>Internet Protocol</i>
IPEA	Instituto de Pesquisa Econômica Aplicada
ITS	<i>Intelligent Transportation Systems</i>
IVI	<i>The Transit Intelligent Vehicle Initiative</i>
JVM	<i>Java Virtual Machine</i>

JSON	<i>JavaScript Object Notation</i>
LGPL	<i>GNU Lesser General Public License</i>
MVC	<i>Model-View-Controller</i>
ODbl	<i>Open Data Commons Open Database License</i>
OSM	<i>OpenStreetMap</i>
OTP	<i>OpenTripPlanner</i>
PBF	<i>Protocolbuffer Binary Format</i>
PNG	<i>Portable Network Graphics</i>
REST	<i>Representational State Transfer</i>
TDM	<i>Transportation Demand Management</i>
TCP	<i>Transmission Control Protocol</i>
TI	Tecnologia da Informação
TICs	Tecnologias da Informação e da Comunicação
TIS	<i>Traveler Information Systems</i>
UFRGS	Universidade Federal do Rio Grande do Sul
UML	<i>Unified Modeling Language</i>
UNISINOS	Universidade do Vale do Rio dos Sinos
URL	<i>Uniform Resource Locator</i>
USIG	Unidad de Sistemas de Información Geográfica
XML	<i>eXtensible Markup Language</i>
WGS	<i>World Geodetic System</i>



## LISTA DE FIGURAS

Figura 2.1: Hierarquia dos Sistemas de Transporte Inteligente (ITS) e dos Sistemas Avançados de Transporte Público (APTS).....	23
Figura 2.2: Página principal do OpenStreetMap ( <a href="http://www.openstreetmap.org">http://www.openstreetmap.org</a> ).....	23
Figura 2.3: Elementos básicos para formar os dados geográficos do OSM. ....	24
Figura 2.4: Estrutura de arquivos de um tile server.....	26
Figura 2.5: Coleta, edição, envio, armazenamento e disponibilização dos dados do OSM. ....	27
Figura 2.6: Conteúdo parcial de alguns arquivos em um feed GTFS.....	28
Figura 2.7: Arquitetura principal do OTP. ....	30
Figura 2.8: Página principal do planejador de viagens da TriMet.....	31
Figura 2.9: Página principal do PoaBus ( <a href="http://www.poabus.com.br/">http://www.poabus.com.br/</a> ).....	32
Figura 2.10: Página principal do serviço GoogleTransit disponível no GoogleMaps ( <a href="https://maps.google.com/transit">https://maps.google.com/transit</a> ).....	33
Figura 2.11: Página de busca do Antares ( <a href="http://antares.unisinos.br/">http://antares.unisinos.br/</a> ).....	35
Figura 2.12: Página de busca do “Mapa Interativo de Buenos Aires” ( <a href="http://www.mapa.buenosaires.gov.ar/">http://www.mapa.buenosaires.gov.ar/</a> ).....	37
Figura 3.1: Diagrama de Casos de Uso. ....	44
Figura 3.2: Interface web - tela de planejamento de viagens. ....	46
Figura 3.3: Interface web - opções adicionais de pesquisa.....	46
Figura 3.4: Interface web - pesquisa textual.....	47
Figura 3.5: Interface web - marcação automática do endereço no mapa e exibição de mensagens de erro e aviso. ....	47
Figura 3.6: Interface web - reiniciar pesquisa e interagir com o mapa. ....	48
Figura 3.7: Interface web - pontos de origem e destino preenchidos.....	49
Figura 3.8: Interface web - listagem dos trajetos encontrados. ....	49
Figura 3.9: Interface web - seleção de trajeto.....	50
Figura 3.10: Interface web - detalhes do trajeto a pé.....	50
Figura 3.11: Interface web - baixar versão mobile.....	51
Figura 3.12: Interface web - seleção de idioma.....	51
Figura 3.13: Interface web - informações do sistema.....	51
Figura 3.14: Interface mobile - tela de planejamento de viagens.....	52
Figura 3.15: Interface mobile - opções adicionais de pesquisa e preenchimento textual dos endereços de origem e destino. ....	53
Figura 3.16: Interface mobile - reiniciar pesquisa, exibição de mensagens e interação com o mapa.....	54
Figura 3.17: Interface mobile - listagem e seleção de trajetos. ....	54
Figura 3.18: Interface mobile – detalhes de trajeto a pé e exibição no mapa.....	55
Figura 3.19: Interface mobile – selecionar idioma e informações do sistema. ....	55

Figura 3.20: Comunicação cliente-servidor. ....	56
Figura 3.21: Modelo cliente-servidor deste trabalho. ....	58
Figura 4.1: Tecnologias empregadas nos servidores e comunicação entre eles. ....	61
Figura 4.2: Tecnologias, comunicação, fonte de dados e estrutura física final dos servidores utilizados pelo ViajeTriFácil. ....	63
Figura 4.3: Interface web - tela de planejamento de viagens. ....	64
Figura 4.4: Interface web – listagem e exibição de trajetos. ....	67
Figura 4.5: Interface web – diferenças em relação ao que foi especificado. ....	68
Figura 4.6: Interface mobile – tela de planejamento de viagens. ....	68
Figura 4.7: Interface mobile – lista e exibição de trajetos. ....	69
Figura 4.8: Geração de aplicações mobile com PhoneGap ( <a href="http://phonegap.com/about">http://phonegap.com/about</a> ). ....	69
Figura 5.1: Teste funcional – Campus Centro UFRGS até PUCRS. ....	72
Figura 5.2: Teste funcional – Campus Centro UFRGS até PUCRS alterando data. ....	72
Figura 5.3: Teste funcional – Campus Centro UFRGS até PUCRS alterando hora. ....	73
Figura 5.4: Teste funcional – Parada de desembarque incorreta no trajeto do Campus Centro UFRGS até Campus Vale UFRGS. ....	73
Figura 5.6: Teste funcional – Shopping Praia de Belas até PUCRS. ....	75
Figura 5.7: Teste funcional – Campus Saúde UFRGS até Estádio Beira Rio. ....	75
Figura 5.8: Teste funcional – Rua Miguel Tostes até Av. Borges de Medeiros considerando 500m de percurso máximo a pé. ....	76
Figura 5.9: Teste funcional – Rua Miguel Tostes até Av. Borges de Medeiros considerando 1km de percurso máximo a pé. ....	76
Figura 5.10: Teste funcional – Av. Bento Gonçalves, 4500, até FIERGS considerando 500m de percurso máximo a pé. ....	77
Figura 5.11: Teste funcional – Av. Bento Gonçalves, 4500, até FIERGS considerando 1km de percurso máximo a pé. ....	77
Figura 5.12: Pesquisa de opinião – perfil do usuário. ....	78
Figura 5.13: Pesquisa de opinião – mobilidade. ....	79
Figura 5.14: Pesquisa de opinião – softwares de mobilidade urbana. ....	80
Figura 5.15: Pesquisa de opinião – avaliação da aparência, facilidade de uso e funcionalidades. ....	81
Figura 5.16: Pesquisa de opinião – avaliação do tempo de resposta e da qualidade das informações apresentadas. ....	82

## LISTA DE TABELAS

Tabela 1.1: Número de veículos na cidade de Porto Alegre na última década .....	15
Tabela 2.1: Áreas ITS e suas principais características.....	20
Tabela 2.2: Categorias APTS e suas principais aplicações .....	22
Tabela 2.3: Comparativo entre as principais funcionalidades das soluções de TIC analisadas.....	38
Tabela 5.1: Comparativo entre as principais funcionalidades de algumas soluções de TIS (Traveler Information Systems) existentes.....	85

## RESUMO

A mobilidade urbana é um dos maiores desafios para as grandes cidades de todo o mundo no início deste século. A opção pelo transporte individual, que no século passado parecia ser a resposta mais eficaz para a mobilidade, deixou de ser sinônimo de autonomia, conforto e qualidade de vida, levando as metrópoles à paralisia do trânsito e, conseqüentemente, causando inúmeros impactos negativos na economia, no ambiente e na sociedade.

Ações que melhorem o gerenciamento e promovam o uso eficiente do transporte precisam ser realizadas para evitar um colapso completo no trânsito e melhorar a mobilidade urbana em grandes cidades como a cidade de Porto Alegre. Nesse contexto, Tecnologias da Informação e Comunicação (TICs) e Sistemas de Transporte Inteligentes (ITS) podem colaborar através da oferta de serviços mais robustos e ágeis que promovam a valorização do transporte coletivo como meio de reduzir o número de veículos nas ruas, incentivando a população a utilizá-lo e deixar o carro em casa.

Com o intuito de estimular e proporcionar um uso mais eficiente e racional do transporte público, o presente trabalho consiste no estudo, na implantação e na validação de um sistema web para facilitar o acesso dos usuários a diferentes pontos da cidade de Porto Alegre através do transporte coletivo. O sistema possibilita aos seus usuários consultar trajetos de ônibus entre dois endereços distintos dentro da cidade de Porto Alegre, fornecendo as melhores opções de trajeto de acordo com as configurações de pesquisa definidas pelo usuário. Ao fornecer os pontos de origem e destino, a solução indicará ao usuário quais opções de ônibus ele poderá utilizar; qual será o tempo médio de espera e de duração da viagem; as integrações, caso seja necessário utilizar mais de um ônibus; e a distância, o tempo e o trajeto para os percursos feitos a pé, se necessário, nos pontos de origem, de destino ou nas integrações.

O sistema desenvolvido neste trabalho faz uso de diversas ferramentas para alcançar uma solução completa, destacando-se a utilização daquelas construídas sobre licenças abertas como o OSM (*OpenStreetMap*), projeto de mapeamento colaborativo que permite a criação, edição e distribuição de dados geográficos do mundo inteiro; OTP (*OpenTripPlanner*), sistema colaborativo que calcula a rota de uma viagem combinando múltiplos meios de transporte; e *Bootstrap, framework* para o desenvolvimento de sistemas web que se adaptam ao tamanho da tela, permitindo que um mesmo site possa ser visualizado tanto em um desktop quanto em um dispositivo móvel.

**Palavras-Chave:** mobilidade urbana, software livre, sistemas de transportes inteligentes, sistemas avançados de transporte público, sistemas de informação ao usuário.

# ViajeTriFácil: A Free Software Based System for Bus Query Paths Applied to the City of Porto Alegre

## ABSTRACT

The urban mobility is a biggest challenges to the large cities around the world in the beginning of this century. The option by individual transport, that in the last century seemed to be the most efficient answer to mobility, is no longer synonymous of autonomy, comfort and life quality, leading metropolises to the paralyzes of traffic and, consequently, causing many negative impacts on the economy, environment and society.

Actions to improve the management and promote the efficient use of transport need to be realized to avoid a complete traffic collapse and to improve the urban mobility in large cities like the city of Porto Alegre. In this context, Information and Communication Technologies (ICT) and Intelligent Transport Systems (ITS) can collaborate by offering more robust and agile services that promote the transit valorization like a way to reduce the number of vehicles on the streets, encouraging the population to use them and to leave the car at home.

Aiming to stimulate and provide a more efficient and rational use of the transit, this work consists in the study, implantation and validation of a web system to facilitate the access of the users to different points of Porto Alegre by transit. The system enable to its users query bus paths between two different addresses in the city, providing the best path options according to the search configurations defined by user. Providing the origin and destination points, the solution will show to user which bus options he can use; what will be the average waiting time and the trip duration; the integrations, if necessary use more than one bus; and the distance, the time and the path for journeys made by feet, if necessary, on origin and destination points or in the integrations.

The system developed in this work makes use of many tools to achieve a complete solution, highlighting the use of those built with open licenses like the OSM (*OpenStreetMap*), a collaborative mapping project that allow the creation, edition and distribution of geographic data of the entire world; OTP (*OpenTripPlanner*), a collaborative system that calculate the route of one trip combining multiples transportation facilities; and *Bootstrap*, framework to the development of web systems that adapt itself to the screen size, allowing that a site can be viewed both on a desktop and on mobile device.

**Keywords:** urban mobility, free software, intelligent transportation systems, advanced public transportation systems, traveler information systems

# 1 INTRODUÇÃO

O transporte exerce um papel de fundamental importância para o desenvolvimento político, social e econômico de um país e uma região, movimentando pessoas, animais e bens de um local para outro. O setor de transportes está em todo lugar gerando empregos, contribuindo para melhorar a distribuição de renda e reduzir a distância entre as zonas rurais e urbanas, melhorando a qualidade de vida da população.

Os serviços de transporte são responsáveis por determinar o acesso da população à educação, à saúde, ao trabalho, ao lazer e a outros serviços, além de proporcionar às empresas a obtenção dos insumos de seus fornecedores e levar seus produtos para os consumidores.

No Brasil, o transporte rodoviário é o principal sistema logístico utilizado, possuindo a maior representatividade entre os modais existentes. O grande desenvolvimento deste meio de transporte no país começou com a industrialização na década de 1930, trazendo a necessidade de maior integração do mercado interno, e se consolidou na década de 1950 com a aceleração da construção de rodovias e a expansão da indústria automobilística, em um momento em que o preço dos combustíveis derivados do petróleo era baixo (MENDONÇA, 2006).

Durante as últimas décadas o transporte individual motorizado (automóveis e motocicletas) foi considerado o meio mais eficaz para dar aos seus usuários maior mobilidade, independência, autonomia e conforto. Nos últimos anos, a taxa de motorização da população brasileira vem apresentando aumento acelerado e os principais fatores que levaram a este crescimento são as vantagens proporcionadas pelo veículo individual, o aumento do poder aquisitivo da população, a concessão de incentivos fiscais concedidos pelo governo e as facilidades na hora da compra (FABICO, 2012-b).

Conforme comunicado do IPEA (2013), no ano de 2012 mais da metade dos lares brasileiros (54%) possuíam veículos privados para seus deslocamentos, sendo o automóvel o principal veículo de posse das famílias, com uma representação de 45% nos lares urbanos. De acordo com o DENATRAN, a frota brasileira de veículos mais que dobrou na última década, passando de 36.658.501, em dezembro de 2003, para 81.600.728, em dezembro de 2013. Na cidade de Porto Alegre, conforme apresenta a Tabela 1.1, apesar do aumento ser menor, a situação não é muito diferente, pois, neste mesmo período a frota passou de 518.351 veículos para 802.932, representando um aumento de aproximadamente 55%. Analisando os dados da tabela, pode-se notar ainda que o principal responsável pelo crescimento no número de veículos na cidade de Porto Alegre são os automóveis, pois representam, para todos os anos analisados, mais de 70% da frota total da cidade. Esta elevada taxa de crescimento não é específica apenas de Porto Alegre, mas sim de grande parte das metrópoles de todo o mundo.

Tabela 1.1: Número de veículos na cidade de Porto Alegre na última década

<i>Ano</i>	<i>Nº Total de Veículos</i>	<i>Automóveis</i>	<i>Ônibus</i>
2003	518.351	403.449	3.316
2005	554.067	424.463	3.452
2007	601.665	450.867	3.817
2009	672.624	495.415	4.161
2011	733.871	529.906	4.690
2013	802.932	571.299	4.970

Fonte: DENATRAN (<http://www.denatran.gov.br/frota.htm>).

O crescimento acelerado e mal planejado da maioria dos centros urbanos traz um grande problema, que é conciliar todos os meios de transporte em um único ambiente. Com o crescimento da população, da maior oferta de carros e do inchaço urbano, o modelo de cidade baseado no transporte individual tornou-se insustentável nas grandes cidades. Atualmente, uma das principais questões nas metrópoles de todo o mundo é como resolver o caos da mobilidade urbana ocasionado por fatores como a falta de planejamento, de investimento público, de infraestrutura para suportar a demanda por espaço viário, aumento da frota de veículos, pouca conscientização da população e pelo ineficiente e caro transporte público. Esses problemas trazem graves consequências políticas, econômicas, sociais e ambientais, tais como o aumento nos congestionamentos, poluição do ar, desperdício de energia fóssil não renovável, perdas econômicas, diminuição do nível de segurança para veículos e pedestres, aumento no tempo de viagem e degradação da qualidade de vida (FABICO, 2012-a).

Se nada for feito, o caos no trânsito vai dificultar ainda mais a vida nas metrópoles nos próximos anos. Esse quadro exige uma nova postura do governo e da sociedade para a busca de soluções. As soluções existem, mas não são simples e devem passar por uma conscientização entre os interesses individuais e coletivos. Algumas medidas já vêm sendo tomadas nos últimos anos em algumas metrópoles do planeta para enfrentar o problema da mobilidade urbana. Londres, por exemplo, desde 2003 aplicam o sistema denominado *congestion charging*, ou taxa de congestionamento, que é a cobrança de uma taxa diária para veículos que entram na região central da cidade, fazendo com que 75 mil veículos deixem de entrar diariamente na região favorecendo o transporte coletivo. Essa mudança possibilitou à cidade colocar mais ônibus em circulação para atender a população e fez com que o tempo de espera caísse quase que pela metade. São Paulo e Cidade do México aplicam o rodízio de automóveis nas ruas baseados no número das placas para restringir a circulação de veículos individuais em determinados dias. Já Curitiba, Cidade do México e Bogotá implantaram o sistema de corredores de ônibus conhecido mundialmente como BRT (*Bus Rapid Transit*), deixando claro que a cidade deve ser construída com foco nas pessoas e não nos automóveis (DISCOVERY, 2012).

As grandes cidades do mundo estão doentes devido aos congestionamentos diários e um dos antídotos é investir, valorizar, qualificar e priorizar o transporte coletivo através de políticas públicas que incentivem sua utilização em detrimento do transporte individual, tais como investimentos na qualidade e na infraestrutura das vias e dos

meios de transporte, maior segurança aos seus usuários, cobrança de tarifas mais justas, incremento na oferta de diferentes modalidades e pontualidade das linhas.

Além das medidas citadas como forma de motivar os usuários a empregar o transporte coletivo, um dos principais fatores que motivam o usuário a empregar tal meio de transporte é a obtenção, de maneira fácil, rápida e precisa, de informação sobre trajetos, horários, integrações e tempo de percurso (JAQUES, 2012). Nesse cenário, a aplicação de estratégias baseadas em TIC (Tecnologias da Informação e Comunicação) para o provimento de ITS (Sistemas de Transporte Inteligente ou *Intelligent Transportation Systems*) torna-se imprescindível para a implantação de cidades inteligentes e podem colaborar de forma efetiva para oferecer serviços mais robustos e ágeis (PASIN, 2014). Sendo assim, é necessário que existam propostas e iniciativas, tanto pela indústria quanto pela academia, na tentativa de criar ferramentas que possam ser acessadas por qualquer dispositivo com acesso à Internet, tais como computadores, tablets e smartphones, auxiliando os usuários no planejamento de suas viagens e fornecendo informações úteis em tempo real. Estas soluções possuem suas vantagens e desvantagens, mas sempre com um propósito em comum que é facilitar o acesso à informação como forma de incentivar e proporcionar um uso mais racional e eficiente do transporte público.

## 1.1 Objetivo

O presente trabalho consiste no estudo, na implantação e na validação de um sistema web, acessível por qualquer dispositivo com acesso à Internet, para possibilitar que seus usuários consultem trajetos de ônibus entre dois endereços distintos dentro da cidade de Porto Alegre, fornecendo as melhores opções de rota de acordo com as configurações de pesquisa definidas pelo usuário. Ao fornecer os pontos de origem e destino, a solução indicará ao usuário quais opções de ônibus ele poderá utilizar; qual será o tempo médio de espera e de duração da viagem; as integrações, caso seja necessário utilizar mais de um ônibus; e a distância, o tempo e o trajeto para os percursos feitos a pé, se necessário, nos pontos de origem, de destino ou nas integrações.

O sistema desenvolvido neste trabalho, denominado de ViajeTriFácil<sup>1</sup>, faz uso de diversas ferramentas para alcançar uma solução completa, destacando-se a utilização daquelas construídas sobre licenças abertas como o OSM (*OpenStreetMap*), projeto de mapeamento colaborativo que permite a criação, edição e distribuição de dados geográficos do mundo inteiro; o OTP (*OpenTripPlanner*), sistema colaborativo que calcula a rota de uma viagem combinando múltiplos meios de transporte; e o *Bootstrap framework* para o desenvolvimento de sistemas web que se adaptam ao tamanho da tela, permitindo que um mesmo site possa ser visualizado tanto em um computador quanto em um dispositivo móvel.

---

<sup>1</sup> <http://mobilidade.inf.ufrgs.br/viajetrifacil>



## 1.2 Organização do trabalho

Este trabalho é constituído por 5 capítulos, além desta introdução. O capítulo 2 apresenta brevemente os Sistemas de Transporte Inteligentes (*Intelligent Transportation Systems* ou ITS), algumas de suas categorias e destaca em qual delas o trabalho proposto se enquadra. Também serão abordadas as iniciativas em software livre OSM (*OpenStreetMap*) para a obtenção de informações de trajetos e mapas e OTP (*OpenTripPlanner*) para o cálculo de rotas entre dois pontos distintos. Além desses tópicos, também será apresentado algumas soluções existentes, suas vantagens e desvantagens em relação a este trabalho. O capítulo 3 fornece a especificação do trabalho proposto discutindo sua arquitetura básica, os principais módulos, interação e interface desejáveis ao usuário. A implementação é mostrada no capítulo 4, abordando as diferentes tecnologias e ferramentas de software usadas no desenvolvimento do trabalho. O capítulo 5 fornece uma avaliação do sistema verificando se ele atende a funcionalidade que se propõe e questões de usabilidade por parte dos usuários finais. Por fim, no capítulo 6, são fornecidas as principais conclusões e resultados deste trabalho, assim como sugestões de melhoria.

## 2 CONTEXTUALIZAÇÃO

O papel da Tecnologia da Informação (TI) no trânsito e nos sistemas de transporte vem passando por uma grande ascensão nos últimos tempos em decorrência dos avanços na tecnologia empregada nos equipamentos eletrônicos e de comunicação, e das melhorias nos sistemas de informação. Esses avanços vêm abrindo novas oportunidades para aprimorar a vazão, segurança e equidade no uso da infraestrutura de transporte existente nos centros urbanos, o qual possui restrições econômicas, ambientais e práticas para que possa se expandir. Esses avanços tecnológicos são a base do que é conhecido hoje como Sistemas de Transporte Inteligente (*Intelligent Transportation Systems* ou ITS) (BAZZAN, 2014, p. 5). Mais especificamente, ITS são programas que envolvem a aplicação e interação de um grupo de tecnologias avançadas, que combinam alta tecnologia e melhorias nos sistemas de informação, nas comunicações, nos sensores e nos métodos computacionais avançados com a gerência e operação dos sistemas de transporte, fazendo com que esses operem com mais segurança, eficiência e mobilidade, além de diminuir os custos sociais, o tempo de espera e de viagem, e o impacto ambiental (SILVA, 2000, p. 4). O objetivo principal dos ITS é unir, de maneira inteligente, os usuários do sistema de transporte, os veículos e a infraestrutura das vias.

### 2.1 Sistemas de Transporte Inteligentes (ITS)

De acordo com (Sussman, 2000), os Sistemas de Transporte Inteligente podem ser classificados em seis diferentes áreas: (i) Sistemas Avançados de Gerenciamento de Tráfego (*Advanced Transportation Management Systems* ou ATMS); (ii) Sistemas Avançados de Informação ao Viajante (*Advanced Traveler Information Systems* ou ATIS); (iii) Sistemas Avançados de Controle Veicular (*Advanced Vehicle Control Systems* ou AVCS); (iv) Operação de Veículos Comerciais (*Commercial Vehicle Operations* ou CVO); (v) Sistemas Avançados de Transporte Público (*Advanced Public Transportation Systems* ou APTS); e (vi) Sistemas Avançados de Transporte Rural (*Advanced Rural Transportation Systems* ou ARTS). A seguir será dada uma breve descrição de cada uma dessas seis áreas com base em (Bazzan, 2014, p. 7), (Silva, 2000, p. 4-5) e (Sussman, 2000, p. 321-326).

O objetivo do ATMS é integrar o gerenciamento de várias funções viárias, tais como os dispositivos de monitoramento de tráfego, as câmeras de vídeo e os semáforos, empregando tecnologias avançadas para prever situações de congestionamento das vias, disponibilizando instruções de trajetos alternativos para melhorar a eficiência da malha viária e garantir maior segurança. Dados em tempo real são coletados, processados e

disseminados pelo ATMS informando os operadores de trânsito sobre rotas alternativas para melhorar a circulação.

Um ATIS é responsável por disponibilizar informações aos viajantes - sejam eles usuários de transporte individual, coletivo ou quaisquer outros participantes do sistema como, por exemplo, pessoas que desejam, antes de iniciar sua viagem, decidir qual modo de transporte utilizar – sobre as condições da via, ambientais e do trânsito, tais como, localização de incidentes, problemas ocasionados por condições climáticas adversas, condições da rodovia, melhores trajetos a percorrer e restrições de tráfego. Tais informações são coletadas e processadas pelos ATMS e então distribuídas para os viajantes através de diversos tipos de mídia com o intuito de garantir maior segurança e reduzir os congestionamentos. Enquanto ATMS se refere à infraestrutura, ATIS está direcionado aos usuários do sistema.

Um AVCS tem a finalidade de melhorar o controle do motorista sobre o veículo, de modo a tornar a viagem mais segura e mais eficiente. Nesses sistemas estão incluídos controles e alertas para prevenir colisões; assistências no controle de direção dos veículos; controle automático lateral e longitudinal, permitindo que veículos possam trafegar mantendo distâncias mínimas entre eles; entre outros.

Na CVO a tecnologia é empregada para aumentar a produtividade, a eficiência e a segurança de veículos comerciais, tais como caminhões, vans e táxis. O monitoramento automático veicular, o gerenciamento de frota, o controle de trajeto e o pagamento eletrônico são exemplos de sistemas destinados a controlar a operação de veículos comerciais.

A área APTS é similar a CVO, representando o uso de tecnologias para aumentar a eficiência, segurança e efetividade, com exceção de que ela é aplicada ao transporte público. Essas tecnologias são utilizadas para beneficiar o usuário dos modais desta categoria - tais como ônibus, trens e metrô - através de aplicações que garantam informações precisas e atualizadas sobre itinerários, trajetos, horários e rotas; facilitem o pagamento da tarifa; minimizem o tempo de espera e de viagem; tragam informações sobre integrações (baldeações) intra e intermodal; utilização da frota; entre outras. Essa área emprega o uso de ATMS, ATIS e AVCS para aumentar a qualidade do serviço e sua eficiência.

Finalmente, o ARTS, cujo objetivo principal é garantir maior segurança, proteção, disponibilidade e movimentação eficiente de veículos que trafegam em áreas rurais, deixando o controle de congestionamentos como objetivo secundário. Exemplos dessa categoria incluem sistemas que fornecem aos motoristas informações sobre condições do tempo e da via, permitindo que eles tomem decisões sobre planos de viagem em longa distância e seleção de rota; sistemas de aviso e alerta com a intenção de reduzir a frequência e a taxa de acidentes; e sistemas de notificação de emergência, para reduzir a exposição do viajante a situações inseguras (perdido ou com o carro quebrado, por exemplo) (U.S. DOT, 1997).

A Tabela 2.1 apresenta um resumo das áreas de ITS e suas principais características.

Tabela 2.1: Áreas ITS e suas principais características

<i>Sigla</i>	<i>Descrição</i>	<i>Características</i>
ATMS	Sistemas Avançados de Gerenciamento de Tráfego	Gerenciamento da rede viária, incluindo gerenciamento de incidentes, controle semafórico, cobrança eletrônica de pedágio, prevenção de congestionamentos e estratégias para contorná-los.
ATIS	Sistemas Avançados de Informação ao Viajante	Fornecer informações aos viajantes antes e durante a viagem em seus veículos. Fornece informações sobre o sistema de transporte em tempo real.
AVCS	Sistemas Avançados de Controle Veicular	Conjunto de tecnologias projetadas para aumentar o controle e a segurança do motorista sobre seu veículo.
CVO	Operação de Veículos Comerciais	Tecnologias que aumentam a produtividade da frota comercial, incluindo localização veicular, procedimentos de pré-liberação, livros de <i>log</i> eletrônico e controle de rota.
APTS	Sistemas Avançados de Transporte Público	Informação e tecnologias disponibilizadas ao passageiro para otimizar a operação do sistema, incluindo coleta de tarifa, transferências intra e intermodal, itinerários, horários, tempo de espera e duração da viagem.
ARTS	Sistemas Avançados de Transporte Rural	Tecnologias para garantir maior segurança e proteção para viagem em áreas pouco povoadas.

Fonte: Adaptado de SUSSMAN, 2000, p. 326

Como este trabalho se destina ao desenvolvimento de uma ferramenta acessível através de qualquer dispositivo com acesso à Internet, possibilitando consultar trajetos, horários, conexões e tempos de espera e de duração da viagem na cidade de Porto Alegre, para disponibilizar ao passageiro de transporte coletivo uma forma de otimizar sua viagem e melhorar sua qualidade de vida, então ele pode ser contextualizado como um Sistema Avançado de Transporte Público (APTS).

## 2.2 Sistemas Avançados de Transporte Público (APTS)

Um APTS procura aplicar a gestão de transporte em conjunto com a tecnologia da informação no sistema de transporte público com o intuito de trazer melhorias na qualidade e na eficiência do serviço prestado à sociedade, proporcionando maior segurança aos seus passageiros, além de oferecer aos usuários maior acesso sobre operações do sistema (U.S DOT, 2000). O objetivo é dar maiores informações aos responsáveis pelo transporte público, sejam eles políticos ou gestores, a fim de possibilitar decisões mais efetivas sobre o sistema, além de aumentar a conveniência e o número de usuários dos modais deste tipo de transporte.

A área APTS pode ser dividida em cinco categorias de acordo com suas relevâncias dentro do transporte público, onde cada uma delas é composta por uma variedade de tecnologias utilizadas para auxiliar agentes de trânsito e organizações a atender as necessidades de serviço dos passageiros. As cinco categorias são: (i) Sistemas de Gerenciamento de Frota (*Fleet Management System* ou FMS), (ii) Sistemas de Informação ao Usuário (*Traveler Information Systems* ou TIS), (iii) Sistemas de Pagamento Eletrônico (*Electronic Payment Systems* ou EPS), (iv) Gestão da Demanda de Transportes (*Transportation Demand Management* ou TDM) e (v) a Iniciativa para Veículo de Transporte Público Inteligente (*The Transit Intelligent Vehicle Initiative* ou IVI).

O objetivo da FMS é impulsionar a eficiência dos sistemas de transporte público, reduzindo os custos de operação e melhorando os serviços através de uma boa precisão em relação à observância da tabela de horários dos coletivos. FMS usa a tecnologia para monitorar a eficiência da frota para atender a demanda dos passageiros, identificar incidentes, gerenciar respostas e restaurar o serviço de forma mais efetiva. O planejamento, a programação e as operações mais eficientes também podem aumentar o número de viagens, fazendo com que mais passageiros utilizem o transporte público como modal preferencial.

Um TIS combina tecnologias de computação e comunicação para fornecer informações do veículo aos passageiros em seu local de trabalho, em sua casa, na rodovia ou em estações de ônibus e/ou trem. As informações permitem aos passageiros escolher o modo de viagem mais eficiente e conveniente. Os viajantes podem acessar horários e rotas em tempo real, reduzir o tempo de espera, consultar itinerários específicos e obter informações sobre congestionamento através de dispositivos móveis, TVs a cabo, painéis eletrônicos ou mesmo através de computadores pessoais. O resultado final é uma maior conveniência no uso e na escolha do transporte desejado para os passageiros rotineiros ou ocasionais.

Os EPS são equipamentos instalados nos modais de transporte público para automatizar o pagamento e a cobrança da tarifa, tornando-a mais segura e conveniente aos passageiros e menos onerosa aos cobradores. Esses sistemas utilizam uma combinação de mídias, tais como cartões inteligentes, com sistemas de comunicação eletrônica, computadores para processamentos de dados e sistemas de armazenamento de dados para coletar as tarifas de forma mais eficiente. Os cartões inteligentes possuem crédito eletrônico e podem ser utilizados em diversos modais de forma integrada, podendo beneficiar o usuário através de descontos na tarifa para quem utiliza mais de uma linha ou outros modais coletivos em um curto período de tempo. Esses sistemas também podem ser utilizados para informar a demanda de viagens em tempo real para um melhor planejamento e programação.

A categoria TDM refere-se a um conjunto de técnicas e programas empregados por agências e organizações de transporte para gerenciar e utilizar a capacidade da infraestrutura existente de forma mais eficiente. O objetivo é maximizar a capacidade da rede de transporte a fim de atender ao aumento da demanda por serviços de transporte. A tecnologia avançada é utilizada para monitorar e gerenciar a capacidade do sistema em tempo real, bem como fornecer alternativas aos viajantes que costumam viajar sozinhos.

Por fim, a categoria IVI é um esforço em pesquisa e desenvolvimento que busca desenvolver tecnologias para evitar acidentes. Os sistemas de segurança e informação

avançados são aplicados para ajudar os motoristas a operar veículos de transporte público de forma mais segura e eficiente.

A Tabela 2.2 apresenta as cinco categorias APTS e suas principais tecnologias.

Tabela 2.2: Categorias APTS e suas principais aplicações

<i>Sigla</i>	<i>Descrição</i>	<i>Principais Tecnologias</i>
FMS	Sistemas de Gerenciamento de Frota	Sistemas de localização automática de veículos, softwares para operações de transporte público, sistemas de comunicações, sistemas de informação geográfica, contadores automáticos de passageiros, sistemas de sinal de trânsito prioritário.
TIS	Sistemas de Informação ao Usuário	Sistemas de informação de trânsito e ao usuário multimodal antes de iniciar a viagem, sistemas de informação de trânsito no terminal e no veículo.
EPS	Sistemas de Pagamento Eletrônico	Cartões inteligentes, sistemas de distribuição de tarifa.
TDM	Gestão da Demanda de Transportes	Sistemas de viagem compartilhada dinâmica, coordenação de serviço automatizado e centros de gerenciamento de transporte.
IVI	A Iniciativa para Veículo de Transporte Público Inteligente	Sistemas para evitar colisão ao mudar e unificar pista, evitar colisão frontal, atenuar impacto da colisão traseira e aumentar precisão de manobras em locais apertados. Ainda está em fase de pesquisa e desenvolvimento.

Fonte: Adaptado de U.S DOT, 2000

A Figura 2.1 ilustra, de forma esquemática, as áreas que compõem os Sistemas de Transporte Inteligentes (ITS), dando especial ênfase aos Sistemas Avançados de Transporte Público (APTS). Dentro deste, salienta-se os Sistemas de Informação aos Usuários (TIS). A razão para esse destaque vem do fato de que o objetivo principal deste trabalho é a implementação de uma ferramenta que forneça informações sobre o transporte público de Porto Alegre aos usuários finais. Dessa forma, o sistema proposto pode ser classificado como um TIS, pertencendo à área APTS.

Nessa linha, existem iniciativas no mundo de software livre que foram concebidas para serem usadas como parte de soluções em ferramentas de Sistemas de Transporte Inteligentes. Essas ferramentas são o *OpenStreetMap* (OSM) para a obtenção de informações sobre trajetos e exibição dos dados geográficos e *OpenTripPlanner* (OTP) para o cálculo das rotas.



Diferentemente das soluções comerciais existentes, onde as empresas decidem o que apresentar no mapa, quais estabelecimentos devem ser exibidos próximo aos locais pesquisados, quais rotas traçar e o que fazer com os dados do usuário, o OSM é uma solução neutra e transparente, onde cada pessoa está livre para utilizar e editar mapas podendo, inclusive, baixar esses mapas e utilizá-los sem estar conectado e sem fornecer sua localização para ninguém.

O *OpenStreetMap* (OSM) está licenciado sob a *Open Data Commons Open Database License* (ODbl), que possibilita a cópia, distribuição, transmissão e adoção dos dados livremente por qualquer um, sendo permitido, inclusive, sua utilização para uso comercial, com a única ressalva de que, ao utilizá-lo, os créditos devem ser atribuídos ao *OpenStreetMap* e seus colaboradores. Além da distribuição de seus dados pela ODbl, o OSM também distribui seus dados cartográficos (imagens) e sua documentação sob a licença *Creative Commons Attribution-ShareAlike 2.0* (CC BY-CA).

Os dados utilizados pelo OSM para a formação do mapa são compostos por quatro elementos básicos: *nós*, que são os pontos usados para marcar locais específicos e para desenhar segmentos de linhas; *caminhos*, que são, basicamente, uma lista de nós conectados, utilizados para criar rodovias, estradas, caminhos, entre outros; *caminhos fechados*, que são caminhos formando um loop completo e são utilizados para criar áreas, tais como parques, lagos, ilhas ou edifícios; e *relações*, que são utilizadas para modelar relacionamentos lógicos, ou geográficos, entre outros elementos, tais como ciclovias, margens de rios, limites administrativos de um país, autoestradas interestaduais e parques nacionais formados pela composição de locais separados fisicamente um do outro. A Figura 2.3 ilustra os elementos do OSM.

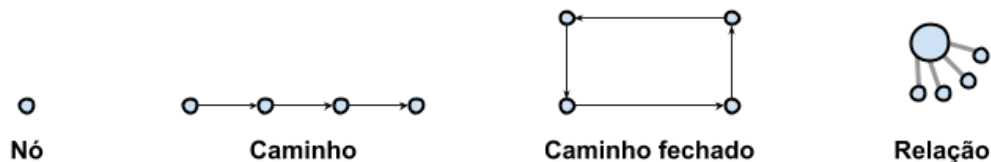


Figura 2.3: Elementos básicos para formar os dados geográficos do OSM.

Cada um dos quatro elementos básicos do OSM pode carregar alguns atributos como nome, tipo e *tag*. O atributo *tag* é composto por uma *chave*, que descreve uma classe de recursos, e um *valor*, que detalha o recurso específico que foi classificado genericamente por uma chave. A *tag* possui formato livre e é utilizada para adicionar atributos não previstos nos elementos. Um exemplo de *tag* é a chave *highway* e o valor *residential*, a qual é atribuída a um elemento do tipo *caminho* para indicar uma rodovia que possui pessoas vivendo ao longo de seu curso. Os principais editores dos dados geográficos do OSM são o iD (ID, 2014), o Potlatch (POTLATCH, 2014) e o JOSM (JOSM, 2014).

Para contribuir com o projeto incluindo novos dados, o usuário precisa: (i) criar uma conta gratuita onde deverá aceitar compartilhar suas contribuições através da licença do *OpenStreetMap*; (ii) coletar os dados, tais como estradas, avenidas, nome de ruas, ou mesmo detalhes mais finos como parques, caixas de correio e cercas vivas, para o OSM através de um GPS, do conhecimento do local, ou mesmo pelo traçado feito por cima das imagens aéreas disponíveis no OSM; (iii) editar o mapa incluindo os elementos básicos que compõem o OSM (nós, caminhos, caminhos fechados ou relações) a partir



das informações coletadas; (iv) incluir detalhes e *tags* a fim de descrever propriedades de um caminho ou nó, tais como seu nome ou seu tipo e, em seguida, submeter as mudanças ao OSM; e, finalmente, (v) visualizar as mudanças feitas no mapa principal do OSM, que serão percebidas alguns minutos após a conclusão da edição, e serão propagadas para os outros servidores que utilizam os mapas baseados nos dados do OSM.

Os principais formatos de dados interpretados pelo OSM são: OSM XML, PBF, O5m e OSM JSON. O OSM XML é um conjunto de arquivos que possuem dados legíveis pelo ser humano contendo uma lista de instâncias dos dados primitivos (nós, caminhos e relações) do modelo OSM e pode ser utilizado pelas principais ferramentas OSM. Esse foi o primeiro formato a ser utilizado pelas APIs (TECMUNDO, 2013) que interagem com o servidor OSM. O PBF é um formato binário, menor para baixar, e muito mais rápido para processar em comparação ao OSM XML, porém nem todas as ferramentas OSM suportam-no e seus dados não são legíveis ao ser humano. O formato O5m foi projetado para ser uma solução intermediária entre o OSM XML e o PBF, tentando combinar as vantagens de ambos através da geração de arquivos de tamanho reduzido, processamento rápido, facilidade na combinação entre dois ou mais arquivos, além de permitir ao usuário escolher seu método de compressão. Por fim, o OSM JSON, uma versão mais leve do OSM XML que utiliza o formato JSON (ECMA, 2013).

O sistema de coordenadas geográficas utilizado pelo *OpenStreetMap* é o WGS-84 (WGS WIKIPEDIA, 2014), o mesmo utilizado pela maioria das unidades de GPS. Contudo, existem muitos outros sistemas que são amplamente utilizados no mundo. Sendo assim, para que o OSM possa utilizá-los, essas coordenadas precisam ser convertidas para o formato WGS-84 para então poderem ser utilizadas.

A exibição dos dados cartográficos do OSM em um mapa se dá através do agrupamento de pequenas imagens denominadas *tiles* que são armazenadas em disco e disponibilizadas para o usuário em *tile servers*, tais como, *CloudMade*, *OpenCycleMap*, *Mapquest*, além do próprio *OpenStreetMap*. A maior diferença entre esses servidores está no estilo em que os dados são exibidos. A ideia não é mostrar a imagem de todo o mapa com uma grande resolução em uma única vez, mas sim exibir apenas a área particular desejada, baixando para a máquina do usuário apenas os *tiles* necessários para a exibição da região requisitada. Dessa forma, o usuário pode navegar na imagem e realizar zoom carregando apenas os dados necessários. O mapa OSM com essas características é denominado *Slippy Map* (SLIPPYMAP, 2013).

Na prática, cada *tile* armazenado no servidor é um arquivo com extensão PNG e possui uma resolução de 256 x 256 pixels. Esses *tiles* são organizados em um sistema de arquivos onde, para cada um dos vinte níveis de zoom existentes, é criado um diretório no *tile server* contendo todas as imagens do mapa para aquele nível específico. Dentro do diretório de cada nível há um conjunto de *tiles* representando o mapa, o qual está distribuído em subdiretórios de modo a estruturar uma matriz de linhas e colunas para facilitar a montagem do mapa.

Cada diretório de um nível específico de zoom ( $n$ ) possui  $2^n$  subdiretórios representando as colunas da matriz de *tiles* e dentro de cada subdiretório os arquivos são numerados em ordem sequencial de 0 até  $2^n$ , de modo a representar as linhas formando todos os *tiles* de cada coluna. Sabendo-se que, para cada nível de zoom ( $n$ ), há  $2^n$  colunas e que em cada coluna há  $2^n$  linhas, então o número total de *tiles* em cada nível de zoom ( $n$ ) é  $2^n * 2^n = 4^n$ . Sendo assim é possível concluir que cada *tile* em um nível

de zoom dá origem a quatro novos *tiles* no nível subsequente a fim de aumentar a escala do mapa de acordo com o acréscimo de zoom. A organização dos *tiles* na estrutura de arquivos do *tile server* é apresentada na Figura 2.4.

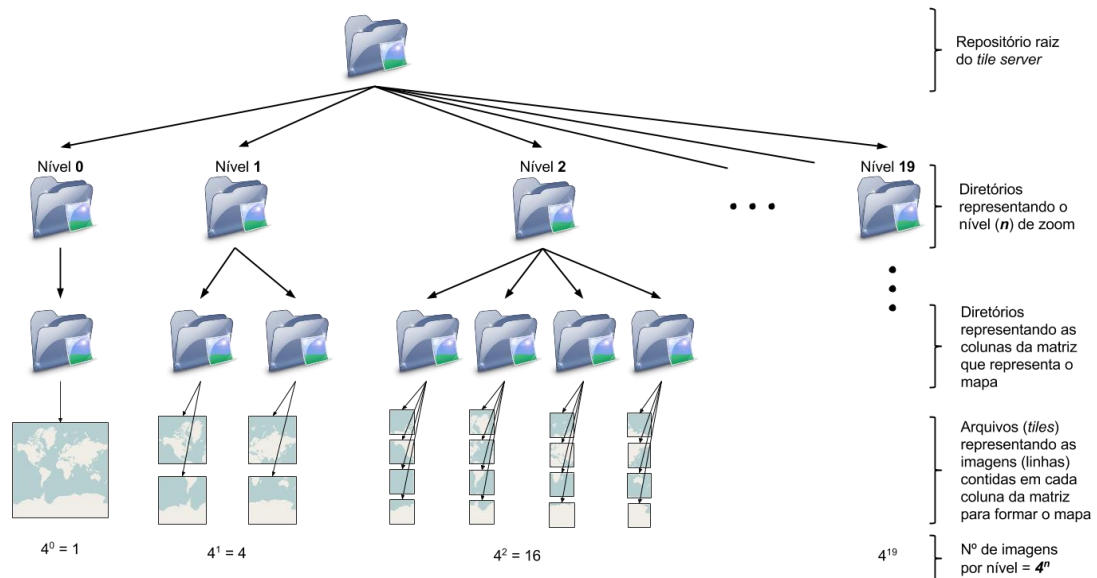


Figura 2.4: Estrutura de arquivos de um *tile server*.

O processo de transformação dos dados do OSM em um conjunto de *tiles* exige um processamento intensivo e não é feito a cada modificação realizada nos dados do OSM, mas sim de tempos em tempos, e os *tiles* gerados são armazenados nos *tile servers*. Caso os servidores existentes não atendam a alguma necessidade específica, ainda é possível criar um *tile server* personalizado (SWITH2OSM, 2013).

Com base no que foi apresentado, é fácil entender como os mapas são gerados e como os dados do OSM são disponibilizados. Primeiramente, informações são obtidas através de GPS (arquivos com extensão GPX), conhecimento local ou traçado sobre imagens aéreas, dados esses que ainda não possuem um formato conhecido pelo OSM. Em seguida, através de um dos editores do OSM (iD, Potlatch e JOSM), o usuário gera um arquivo contendo os dados em um dos formatos conhecidos pelo OSM. Na sequência o arquivo é submetido ao servidor OSM e os dados são manipulados internamente pelas ferramentas *osm2pgsql* (OSM2PGSQL, 2014) e *osmosis* (OSMOSIS, 2014), as quais realizam a conversão dos dados para o formato PostGIS (POSTGIS, 2014) e armazenam-nos no banco de dados PostgreSQL do servidor OSM. Por fim, os dados geoespaciais armazenados no servidor OSM são disponibilizados de diferentes formas a fim de fornecer informações e funcionalidades úteis para beneficiar a comunidade, tais como: disponibilização dos dados do mapa para download, mapas cartográficos (*Slippy Maps*), busca de dados OSM por nome e endereço e planos de rotas.

Os dados do OSM são disponibilizados para download através de ferramentas como *osm2pgsql*, *pbftosm* (PBFTOOSM, 2011), *osmconvert* (OSMCONVERT, 2014), ou mesmo através das APIs *OSM Editing API* (OSMAPI, 2013), *XAPI* (XAPI, 2014) e *Overpass API* (OSM3S, 2014), em um dos formatos OSM conhecidos. Os mapas cartográficos (*Slippy Maps*) são gerados através da ferramenta *Mapnik* (MAPNIK, 2013) - a qual é responsável por gerar as imagens (*tiles*) a partir dos dados geoespaciais do servidor OSM – e são disponibilizados em *tile servers*. A busca de dados OSM por

nome e endereço, e geração de dados sintéticos a partir de coordenadas (geocodificação reversa) é feita através da ferramenta *Nominatim* (NOMINATIM, 2014). E, por fim, os planos de rota realizados para ajudar pessoas a irem de um lugar até outro são feitos através de ferramentas como *OpenTripPlanner* (OTP, 2014) e *Navit* (NAVIT, 2014). A Figura 2.5 ilustra os passos para a geração e disponibilização dos dados do OSM.

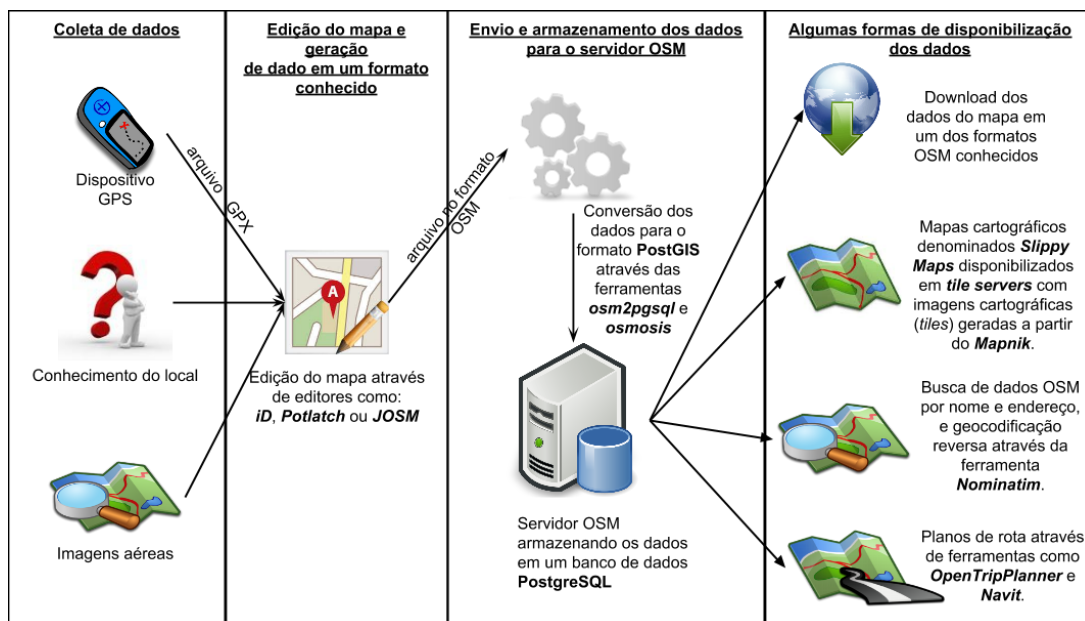


Figura 2.5: Coleta, edição, envio, armazenamento e disponibilização dos dados do OSM.

## 2.4 OpenTripPlanner (OTP)

*OpenTripPlanner* (OTP) é um projeto de código aberto criado em 2009 pela agência de transportes TriMet na cidade de Portland, OR, USA. O sistema é utilizado para planejar viagens multimodais e para analisar redes de transporte, fornecendo uma interface web e uma API para interação com outras aplicações, possibilitando ao usuário planejar rotas utilizando vários “modos” como, por exemplo, carro, bicicleta, transporte público, a pé, ou mesmo uma combinação deles, permitindo adaptar a viagem de acordo com as preferências pré-estabelecidas. A plataforma é mantida e atualizada através de um esforço colaborativo entre iniciativas individuais, agências de trânsito e organizações sem fins lucrativos que objetivam integrar o projeto com algumas iniciativas de código aberto e, ao mesmo tempo, desenvolver uma ferramenta fácil de usar, flexível, confiável e rápida (FAQ OTP, 2013).

O OTP é disponibilizado sob a versão 3 da *GNU Lesser General Public License* (LGPL), uma licença de software livre a qual visa à regulamentação no uso de bibliotecas de código, mas também pode ser aplicada na regulamentação de aplicações (LGPL, 2007). A principal característica dessa licença é a liberdade de associar softwares licenciados através dela com outros programas regulamentados por licenças livres como GPL (GPL, 2007) e LGPL, ou mesmo com programas proprietários.

Para construir a rede de transporte multimodal utilizada para identificar e traçar os caminhos, o OTP opera sobre um objeto denominado *grafo*, o qual possui todos os locais da região mapeada, os caminhos para ir de um ponto até outro e os horários e

itinerários dos meios de transporte disponíveis (GRAPH OTP, 2014). Para construí-lo, o OTP utiliza arquivos no padrão de dados abertos, ou seja, constrói o grafo a partir de dados disponibilizados livremente para uso e republicação, sem restrições de direitos autorais.

A construção da rede de caminhos utilizada para definir as rotas no grafo é feita pelo OTP a partir de dados obtidos do OSM ou através de *shapefiles* (ESRI, 1998). Os dados do OSM podem ser utilizados através de arquivos no formato OSM XML previamente gerados, ou mesmo através de uma integração direta do OTP com uma das APIs do OSM disponíveis, forma esta que fará com que os dados do OSM sejam importados apenas no momento da construção do grafo. A outra forma é através de arquivos no formato *shapefile*, os quais possuem dados espaciais representados na forma de vetores.

Os dados que descrevem os horários e as informações geográficas dos transportes públicos no grafo são obtidos através de arquivos definidos pela Especificação Geral de Feeds de Transporte Público (*General Transit Feed Specification* ou GTFS) que, normalmente, são gerados e disponibilizados pelas agências de transporte público. Cada *feed* GTFS é composto por um conjunto de arquivos de texto, onde cada um deles modela aspectos específicos sobre o transporte público, tais como paradas, trajetos, viagens, horários e outros dados relevantes, que são armazenados em um arquivo único compactado no formato ZIP (GOOGLE TRANSIT, 2013). A Figura 2.6 ilustra o conteúdo parcial existente em dois arquivos que fazem parte de um *feed* GTFS disponibilizado pela agência de transporte público da cidade de Porto Alegre (EPTC), onde o primeiro contém informações relativas aos pontos de parada e o segundo contém informações das rotas percorridas pelos veículos.

#### stops.txt

```
stop_id,stop_code,stop_name,stop_desc,stop_lat,stop_lon
4953,,End.: FRANCISCO TALAIA DE MOURA DF 345,Tipo de abrigo: Piso: 250/13",-30.00291,-51.090739
2861,,End.: MONTENEGRO 235,Tipo de abrigo: Piso: "S1",-30.040881,-51.181563
3857,,End.: FRANCISCO TREIN DF 326,Tipo de abrigo: Piso: "CEEE",-30.013295,-51.159602
3856,,End.: FRANCISCO TREIN AL 326,Tipo de abrigo: Piso: "S1",-30.013102,-51.159545
3123,,End.: FRANCISCO TREIN DF 707,Tipo de abrigo: Piso: "S1",-30.017333,-51.158982
3851,,End.: UMBU 701,Tipo de abrigo: Piso: "S1",-30.015224,-51.162233
5031,,End.: LUIZA ROCCO DF 200,Tipo de abrigo: Piso: , -30.073109,-51.203312
2471,,End.: BARCELLOS 1706,Tipo de abrigo: Piso: "S1",-30.088822,-51.180341
```

#### routes.txt

```
route_id,agency_id,route_short_name,route_long_name,route_desc,route_type,route_url,route_color,route_color2
343-2,EPTC,343,CAMPUS / IPIRANGA-2,,3,,CFD600,000000
```

Figura 2.6: Conteúdo parcial de alguns arquivos em um *feed* GTFS.

O OTP também pode utilizar dados de elevação de terreno no formato GeoTIFF (TRAC, 2009) para complementar o grafo e utilizá-los para criar rotas de bicicleta levando em conta as inclinações do trajeto percorrido.

Com o grafo gerado, o OTP está apto a realizar o cálculo dos caminhos entre pontos distintos através de um dos quatro algoritmos disponíveis a fim de definir as melhores opções de rota de acordo com configurações pré-estabelecidas. Os primeiros algoritmos de roteamento utilizados pelo OTP foram o *Dijkstra* (DIJKSTRA, 1959) e o *A\** (HART, 1968) utilizando como heurística a *distância Euclidiana*. Esses métodos eram

relativamente fáceis de implementar e apresentavam um bom desempenho para grafos pequenos e médios. Contudo, ao utilizar grafos maiores, o desempenho apresentado por esses algoritmos não era satisfatório (OTP OSM, 2014). A partir desse problema, a comunidade OTP passou a utilizar a técnica denominada *Contraction Hierarchies* (GEISBERGER, 2008), a qual é baseada na ideia de que os grafos podem ser contraídos através da remoção de vértices e substituição dos caminhos através destes por “atalhos”, o que deveria apresentar um desempenho melhor para grafos maiores. Contudo, após aplicar essa técnica, o desempenho não foi tão satisfatório quanto o esperado, pois essa técnica não foi projetada para levar em conta nodos de transporte público que são variáveis com o tempo (OTP OSM, 2014). Logo, essa solução passou a ser empregada apenas em segmentos do grafo que não utilizam transporte público. A solução mais empregada atualmente é a utilização dos algoritmos A\* com distância Euclidiana ou *Contraction Hierarchies* para viagens a pé ou de bicicleta. Para viagens que combinam esses modos com o transporte público são utilizadas variações do algoritmo A\* para múltiplos objetivos com melhorias no desempenho do roteamento através do emprego de heurísticas específicas (ROUTING OTP, 2014).

Como alternativa às técnicas de roteamento existentes, ainda existe o algoritmo *RAPTOR* (DELLING, 2012), o qual é projetado especificamente para roteamento em redes de transporte público e possui desempenho melhor em grafos maiores (RAPTOR OTP, 2013). Diferentemente dos algoritmos anteriores, o *RAPTOR* não é baseado no algoritmo de Dijkstra. Basicamente, ele analisa cada rota, como, por exemplo, uma linha de ônibus, no máximo uma vez a cada iteração e pode fazê-la mais rápido aplicando simples regras de poda e processamento paralelo. Apesar de possuir bom desempenho em redes maiores, seu desempenho é inferior em comparação aos demais para grafos pequenos. Portanto, para compensar esse problema, ao utilizá-lo, o OTP mantém a aplicação dos algoritmos de Dijkstra ou A\* para rotas que não possuem transporte público (RAPTOR OTP, 2013).

A fim de disponibilizar as funcionalidades do OTP, como as opções encontradas pelo algoritmo de roteamento para encontrar os melhores caminhos sobre o grafo e os itinerários existentes, o servidor OTP disponibiliza, através de um servidor web Java, um serviço web baseado no estilo arquitetural REST (FIELDING, 2000). Para sistemas de hipermídia, como a *World Wide Web*, o servidor recebe e responde a requisições HTTP através de dados nos formatos XML ou JSON (OTP, 2014). Por estar baseado nessa arquitetura, esse serviço web é chamado de uma API RESTful.

O projeto OTP é composto basicamente por quatro módulos de software principais: *GraphBuilder*, *Routing*, *OTP Routing API* e *Primary UI*. O *GraphBuilder* é o utilitário de linha de comando utilizado por realizar a transformação da rede de caminhos (OSM ou *shapefiles*), dos horários e informações de transportes públicos (GTFS) e dos dados de elevação de terreno (GeoTIFF) em um *grafo*. O *Routing* é a biblioteca núcleo do OTP, responsável por calcular a árvore de caminhos mais curtos entre dois pontos distintos no *grafo* através de um dos algoritmos de roteamento disponíveis. O *OTP Routing API* é o serviço web integrado com o *Routing* a fim de fornecer os itinerários encontrados a partir de requisições específicas feitas por outras aplicações empregando a arquitetura REST. E, por fim, o *Primary UI* é o sistema web disponibilizado pelo OTP, integrado com o *OTP Routing API*, responsável por disponibilizar a representação visual de um mapa e das rotas encontradas a partir de pesquisas específicas feitas pelo usuário. A Figura 2.7 ilustra a arquitetura de um servidor OTP apresentando seus principais módulos e a comunicação entre eles.

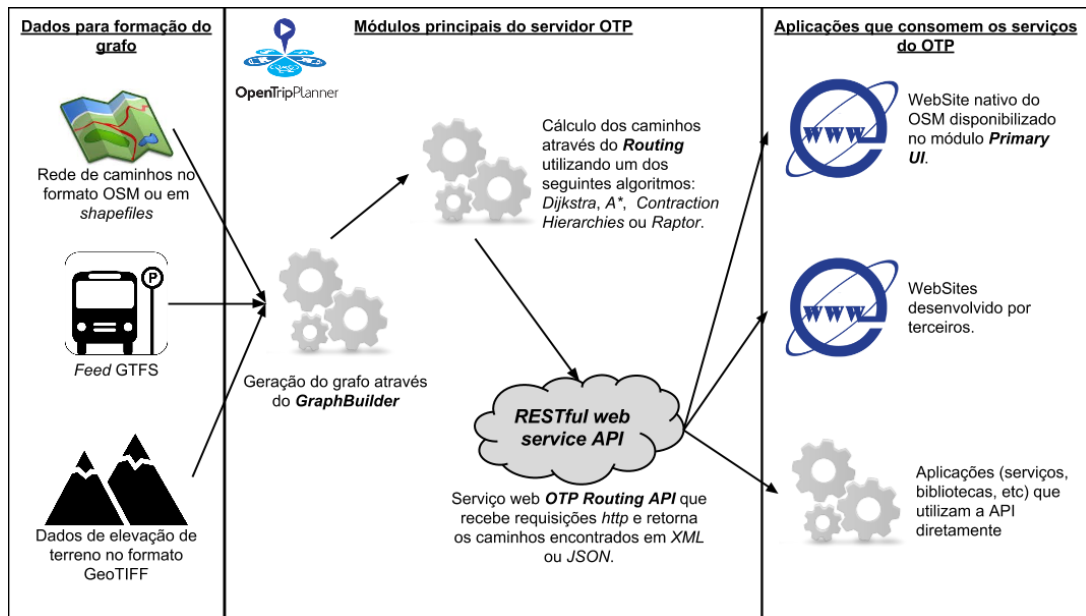


Figura 2.7: Arquitetura principal do OTP.

A linguagem de programação principal empregada no desenvolvimento dos módulos do OTP é Java, com exceção do *Primary UI*, que foi desenvolvido em JavaScript.

A instalação de um servidor OTP pode ser realizada tanto em máquinas com sistema operacional Windows quanto GNU/Linux, ou mesmo em outras plataformas, pois a linguagem de programação principal empregada no projeto é o Java, a qual não é compilada em código nativo. Portanto, para disponibilizar a API do OTP é preciso um servidor contendo uma plataforma que suporte uma máquina virtual Java (*Java Virtual Machine* ou JVM), um servidor web com suporte a conteúdos dinâmicos usando a plataforma Java, tais como os aplicativos Tomcat (TOMCAT, 2014) ou o Jetty (JETTY, 2014) e, por fim, os módulos OTP e suas dependências.

Originalmente utilizado pela agência de transportes TriMet, o OTP já está em vários lugares do mundo e é também o mecanismo de roteamento por trás de vários aplicativos para smartphones. Estes são alguns dos sistemas disponíveis na web que utilizam OTP como seu mecanismo de definição de rotas: *TriMet Map Trip Planner*<sup>2</sup>, Portland, Oregon, USA; *EMT Valencia*<sup>3</sup>, Valencia, Spain; *Adelaide Metro*<sup>4</sup>, Adelaide, Australia; *Moovit*<sup>5</sup>, Tel Aviv, Israel (DEMOS OTP, 2014). A Figura 2.8 apresenta a página principal do sistema da TriMet, disponibilizado em Portland, que utiliza um servidor OTP em combinação com o OSM para o planejamento das rotas de transporte público.

<sup>2</sup> <http://ride.trimet.org/#/>

<sup>3</sup> [http://www.emtvalencia.es/geoportal/?lang=en\\_otp](http://www.emtvalencia.es/geoportal/?lang=en_otp)

<sup>4</sup> <http://jp.adelaidemetro.com.au/opentripplanner-webapp/#/>

<sup>5</sup> <http://tripplan.moovitapp.com/>

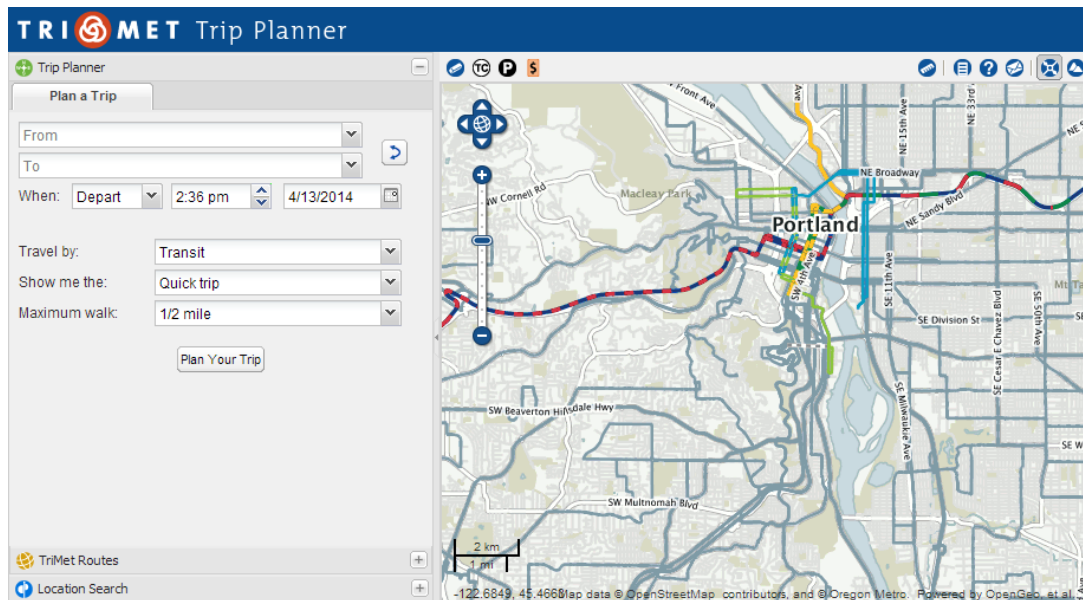


Figura 2.8: Página principal do planejador de viagens da TriMet.

## 2.5 Sistemas de Mobilidade Urbana Existentes

Soluções baseadas no *OpenStreetMap* (OSM) e no *OpenTripPlanner* (OTP) para criar aplicações com o objetivo de facilitar a vida de usuários de transporte público não são únicas e, assim como essas, existem diversas outras, sejam por iniciativas acadêmicas, corporativas ou mesmo individuais, que já estão sendo empregadas em diversas cidades de todo o mundo. Esta seção apresenta algumas das soluções que estão sendo empregadas em Porto Alegre e em outras cidades do Brasil e do mundo.

Inspirado em soluções já existentes em outras cidades, Bruno Jurkowski, estudante de Ciência da Computação na UFRGS, criou, no início de 2011, por iniciativa própria, o *PoaBus*<sup>6</sup>, um sistema web pelo qual é possível descobrir qual ônibus faz determinado itinerário, em quais paradas ele faz embarque e desembarque, e quais as linhas pegar para ir de um endereço até outro na cidade de Porto Alegre. O sistema, com sua tela principal ilustrada na Figura 2.9, além de ser uma iniciativa pioneira aplicada ao transporte público em Porto Alegre, é gratuito, fácil de usar e também possui a possibilidade da inclusão de rotas por parte dos internautas de forma colaborativa. As principais desvantagens dessa ferramenta se devem ao fato de não considerar as integrações na busca de trajetos, não possuir informações de horários de partida e nem tempo de viagem dos ônibus, não possuir outros modais de transporte, além de ônibus, não possuir versão para dispositivos móveis e ser um software proprietário em que o código fonte da aplicação não é compartilhado com o público.

<sup>6</sup> <http://www.poabus.com.br>

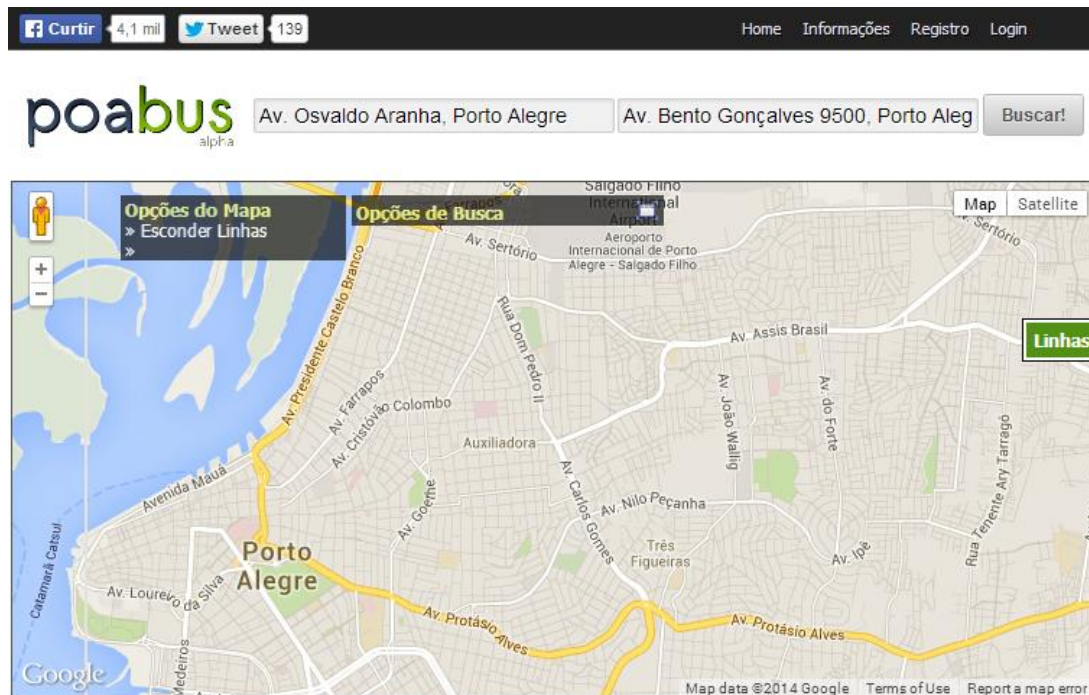


Figura 2.9: Página principal do *PoaBus* (<http://www.poabus.com.br/>).

Dez meses após a criação do *PoaBus*, em dezembro de 2011, o *PoaTransporte*<sup>7</sup> foi lançado através de uma parceria entre a prefeitura de Porto Alegre, a EPTC e a PROCEMPA a fim de fornecer uma solução web por parte da iniciativa pública para a mobilidade urbana na cidade de Porto Alegre. A solução possibilita a pesquisa de ônibus e lotações por código, nome, endereço ou parada associada a linha. Entretanto, essa ferramenta não atendeu as expectativas e as necessidades da maioria dos usuários por não oferecer funcionalidades básicas como a consulta de trajetos entre dois endereços distintos, não possuir informações sobre outros modais de transporte, entre outras.

O *Google Transit*<sup>8</sup> é um recurso disponível no serviço de visualização e pesquisa de mapas *Google Maps*, que já está disponível para cerca de 800 cidades de todo o mundo, pelo qual o usuário pode consultar as linhas de ônibus mais próximas, os trajetos possíveis, o tempo médio de espera, o tempo de viagem e também as integrações entre diferentes linhas de ônibus, ou mesmo entre diferentes modos de transporte. O serviço é de fácil utilização, gratuito e pode ser acessado por qualquer dispositivo com acesso à Internet através do sistema web ou mesmo através do aplicativo disponível para smartphones e tablets. Mas, apesar de ser gratuito, o *Google Transit* é um software proprietário, suprimindo o conhecimento em torno do programa e a inovação a partir dele. Isso faz com que seja impossível saber quais as estratégias utilizadas para definir os locais e os trajetos que a ferramenta indica. Outra desvantagem da ferramenta é que, apesar de ser empregada em muitas cidades do mundo, inclusive em algumas cidades do Brasil como São Paulo, Rio de Janeiro, Curitiba e Porto Alegre, ela dificilmente será expandida para cidades menores, seja pela dificuldade em implantar uma solução global como essa em grande parte das cidades de todo o mundo ou mesmo pelo baixo retorno

<sup>7</sup> <http://www.poatransporte.com.br>

<sup>8</sup> <https://maps.google.com/transit>



econômico que traria. A ferramenta também não leva em consideração alguns modais de transporte rodoviários e nem hidroviários em algumas cidades em que ela está implantada, além de não considerar informações do trânsito em tempo real para a definição das rotas de transporte público. Outro detalhe, menos significativo, é que a ferramenta exibe apenas o código das linhas de ônibus pesquisadas, não apresentando o nome de cada uma. Na cidade de Porto Alegre, por exemplo, essa informação é muito importante para que as pessoas consigam identificar qual ônibus pegar. A Figura 2.10 apresenta a página principal do *Google Maps* contendo o resultado de uma pesquisa realizada através do serviço *Google Transit*.

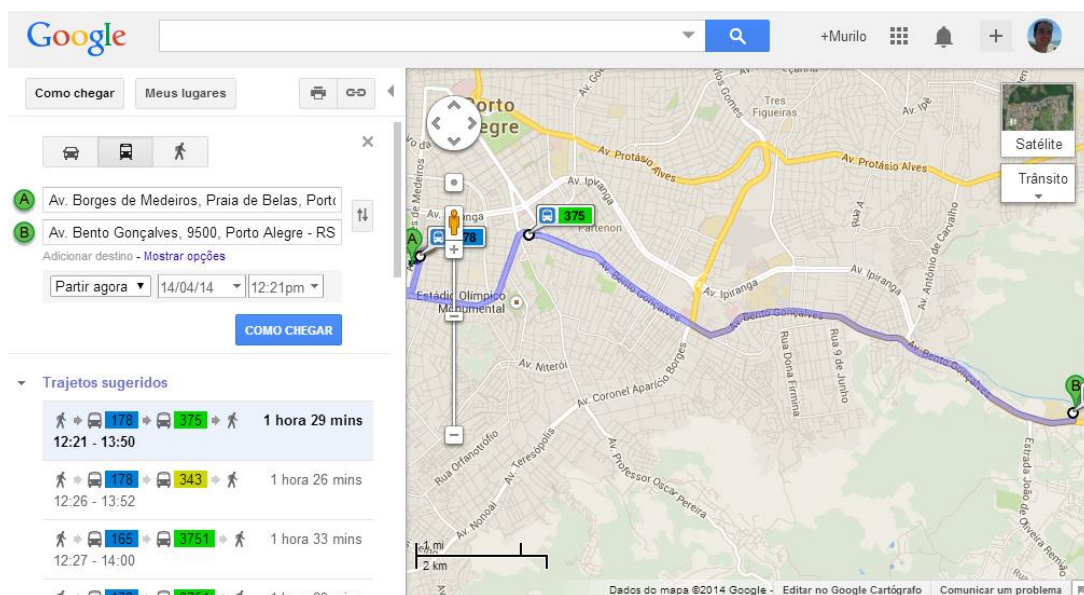


Figura 2.10: Página principal do serviço *GoogleTransit* disponível no *GoogleMaps* (<https://maps.google.com/transit>).

O *Moovit*<sup>9</sup> é uma solução gratuita que pode ser instalada em dispositivos móveis que possuam sistemas Android ou iOS ou mesmo acessada via web. A ferramenta foi projetada com o objetivo de fornecer as rotas dos usuários até seus destinos desejados baseado em informações em tempo real. A solução já está disponível em diversas cidades do mundo, inclusive em cidades do Brasil como São Paulo, Rio de Janeiro, Belo Horizonte e Porto Alegre. Dentre os grandes diferenciais da ferramenta estão: facilidade de uso; combinação de dados de transporte com os dados em tempo real gerados pelos usuários de forma anônima (localização e velocidade do veículo) ou ativa (relatórios), fornecendo uma visão mais precisa sobre os tempos de viagem e de espera; utilização do *OpenStreetMap* (OSM) para exibir os mapas e obter informações das ruas; integração com dispositivo GPS para emissão de alertas ao usuário informando em qual parada descer; e cálculo de trajetos levando em conta todos os meios de transporte disponíveis como ônibus, metrô e trem. Por ser uma solução corporativa e global, as desvantagens apresentadas por ela são similares àquelas apresentadas pelo *Google Transit*, pois seu código fonte é proprietário, não está disponível para muitas cidades, exibe apenas o código das linhas e, além disso, faltam informações sobre empresas de transporte público nas cidades disponíveis, demanda tempo de busca por trajetos e exibe propagandas publicitárias.

<sup>9</sup> <http://www.moovitapp.com>

Outra solução para auxiliar usuários de ônibus na cidade de Porto Alegre é o *NossoBus*<sup>10</sup>, um aplicativo desenvolvido por 4 estudantes universitários disponível gratuitamente para dispositivos móveis com plataforma Android. Através dele o usuário informa apenas o local que deseja chegar e o aplicativo, utilizando informações da rede de dados 3G e de geolocalização através do GPS, identifica a localização atual do usuário, indica a parada de ônibus mais próxima, qual ônibus pegar, acompanha no mapa o deslocamento do usuário, indicando em qual parada descer e onde está o destino final. O aplicativo é muito fácil e prático de utilizar e possui uma integração interessante com o GPS do smartphone, guiando o usuário até seu destino final. Contudo, o sistema apresenta algumas limitações, entre elas: não possibilita a consulta de uma rota a partir de um local diverso daquele em que o aparelho está situado; não exibe e nem possibilita a escolha de diferentes opções para um mesmo trajeto; não possui versão para web e nem para outras plataformas móveis; não dá suporte a integrações; é aplicado apenas para a cidade de Porto Alegre; e, além disso, as mensagens apresentadas pela aplicação não estão no mesmo idioma, com algumas sendo exibidas em português e outras em inglês.

Um outro sistema web, existente desde 2008, é o *Antares*<sup>11</sup>, desenvolvido na Universidade do Vale do Rio dos Sinos (UNISINOS). Através dele o usuário fornece os locais de origem e destino e o sistema sugere até cinco rotas possíveis informando, para cada uma delas, os caminhos, seja a pé ou de ônibus, de forma textual e no mapa, as integrações e o horário de partida. A aplicação é gratuita, utiliza como sistema de roteamento o *OpenTripPlanner* (OTP), possui um visual atraente e uma boa usabilidade, apresenta os resultados de busca muito rapidamente e está disponível em dois idiomas, português e inglês. Os pontos desfavoráveis da ferramenta ficam por conta de: não haver versões para aplicativos móveis; não ser responsivo, ou seja, não apresentar as informações de forma acessível e confortável para diferentes meios de acesso como, por exemplo, telas de diferentes tamanhos; não exibir outras opções de busca como, por exemplo, a possibilidade de informar o trajeto máximo desejado a pé e o horário de partida ou chegada; não exibir detalhes em relação aos trajetos a pé; não informar ao usuário o tempo de duração da viagem; e estar disponível apenas para a cidade de Porto Alegre. A Figura 2.11 mostra a página de busca por trajetos do projeto *Antares*.

---

<sup>10</sup> <https://play.google.com/store/apps/details?id=com.izee.nossobus>

<sup>11</sup> <http://antares.unisinos.br>

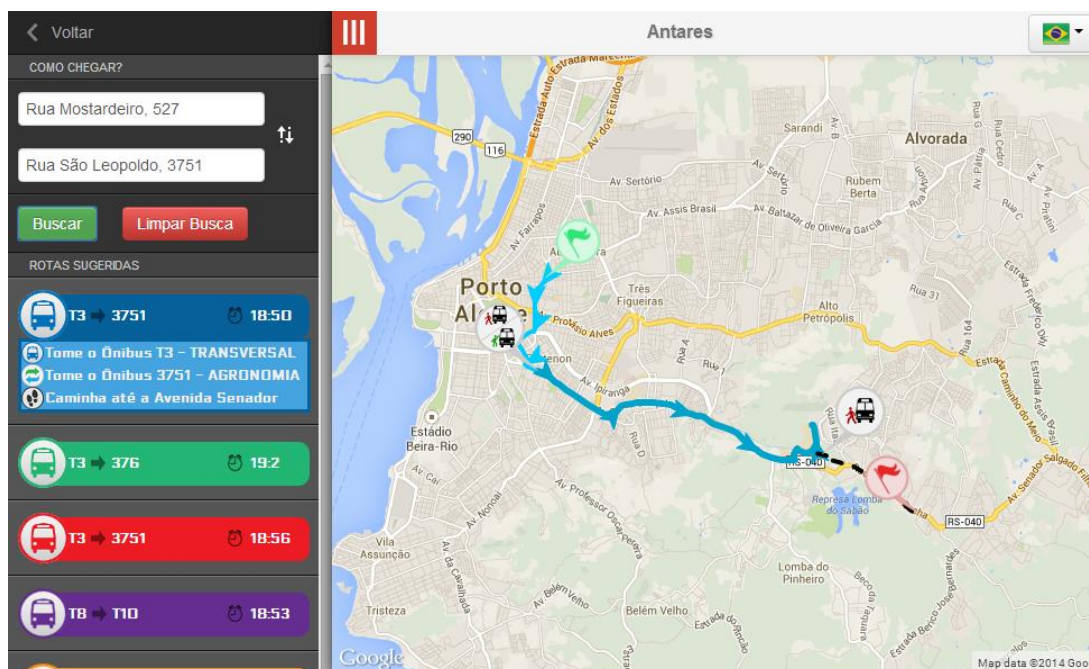


Figura 2.11: Página de busca do Antares (<http://antares.unisinos.br/>).

Outra iniciativa muito inovadora, desenvolvida por um grupo de pessoas denominado *Shoot The Shit*, foi a criação e execução do projeto “*Que Ônibus Passa Aqui?*”<sup>12</sup>. Diferentemente das demais soluções apresentadas nessa seção, essa não se trata de uma ferramenta computacional, mas sim de um projeto que busca sinalizar de forma colaborativa os pontos de ônibus das cidades brasileiras imprimindo e colando adesivos nas paradas que, inclusive, estão disponíveis para *download* no site do projeto, a fim de dar a possibilidade para os usuários preencherem no próprio adesivo as linhas que cruzam pela parada. O projeto, na verdade, trata-se de uma espécie de alerta para às autoridades enxergarem esse grave problema que é a falta de informação sobre linhas de ônibus em cidades como Porto Alegre, que foi onde o projeto iniciou. Após a grande receptividade do público frente a essa iniciativa, o projeto chamou a atenção da mídia e acabou sendo apoiado pela prefeitura de Porto Alegre que, através de uma parceria com os criadores do projeto, desenvolveu um adesivo oficial que foi colado em inúmeras paradas da cidade. Contudo, devido ao fato dos adesivos estarem sendo arrancados pela própria população, o projeto acabou sendo interrompido, mas, mesmo assim, a iniciativa foi tão interessante que o projeto acabou sendo reconhecido em todo o Brasil.

Outra solução tecnológica interessante desenvolvida para auxiliar os usuários de transporte público de Porto Alegre é o *Moobly*<sup>13</sup>, um aplicativo gratuito para dispositivos com sistema Android que possui uma interface prática e bem informativa, na qual é possível: consultar informações tarifárias de lotações, ônibus, táxi, bicicletas e trem; consultar trajetos entre dois pontos distintos utilizando ônibus ou táxi; descobrir as estações de bicicleta, trem, ônibus e táxi próximas de um local específico; consultar horários, tarifas e itinerários das linhas de ônibus; salvar as linhas de ônibus em uma lista de favoritos; selecionar um ponto no mapa através de geolocalização GPS, seleção textual ou mesmo através de marcação no mapa; e também chamar viaturas de táxi pelo

<sup>12</sup> <http://www.shoottheshit.cc/Que-Onibus-Passa-Aqui>

<sup>13</sup> <http://moobly.com.br/>

próprio aplicativo. A ferramenta possui uma restrição em relação ao cálculo de trajetos, pois ela apenas lista as paradas próximas do usuário que possuem um meio de transporte em comum entre pontos de origem e destino pesquisados, não exibindo no mapa o caminho percorrido pelo modal. Além disso, a ferramenta não considera possíveis integrações ou transferências multimodais no cálculo dos trajetos; é um software proprietário; a consulta de linhas e trajetos é lenta; não possui versões para outras plataformas; e está disponível apenas para a cidade de Porto Alegre.

Por fim, uma solução que vale a pena ser destacada é o “*Mapa Interativo de Buenos Aires*”<sup>14</sup>, um software gratuito desenvolvido utilizando exclusivamente software livre, tanto no que diz respeito aos componentes e os serviços que a formam, quanto às ferramentas de desenvolvimento utilizadas para sua construção, que foi desenvolvido pela *Unidad de Sistemas de Información Geográfica (USIG)*, órgão pertencente ao gabinete de governo eletrônico da cidade de Buenos Aires, Argentina, e tem o objetivo principal de disponibilizar uma solução para planejar viagens dentro da cidade levando em conta os meios de transporte público disponíveis. A ferramenta pode ser acessada por qualquer dispositivo através de um navegador web com acesso à Internet, visto que há versões para estações de trabalho e dispositivos móveis, ou através de aplicativos disponíveis para Android e iOS. A ferramenta se destaca por ser uma ótima solução implantada pela iniciativa governamental; por utilizar software livre para sua construção; por permitir a busca de trajetos entre endereços distintos levando em conta todos os meios de transporte público disponíveis na cidade; por calcular integrações e transferências intermodais; por disponibilizar inúmeras informações de interesse social no mapa como, por exemplo, a divisão dos bairros da cidade, a localização de bancos, ciclovias, hospitais, farmácias e escolas, e a exibição de imagens aéreas da cidade ao longo do tempo; além de fornecer informações detalhadas sobre cada ponto da cidade. Os inconvenientes da ferramenta ficam por conta dela não estar disponível para outras cidades, apresentar relativa lentidão para apresentar o resultado das buscas por trajetos, não estar disponível em outros idiomas além do espanhol, não permitir a seleção de locais diretamente no mapa e não exibir informações detalhadas sobre o trajeto que deve ser percorrido a pé. A Figura 2.12 apresenta o resultado de uma pesquisa feita entre dois endereços distintos no “*Mapa Interativo de Buenos Aires*”.

---

<sup>14</sup> <http://www.mapa.buenosaires.gov.ar>

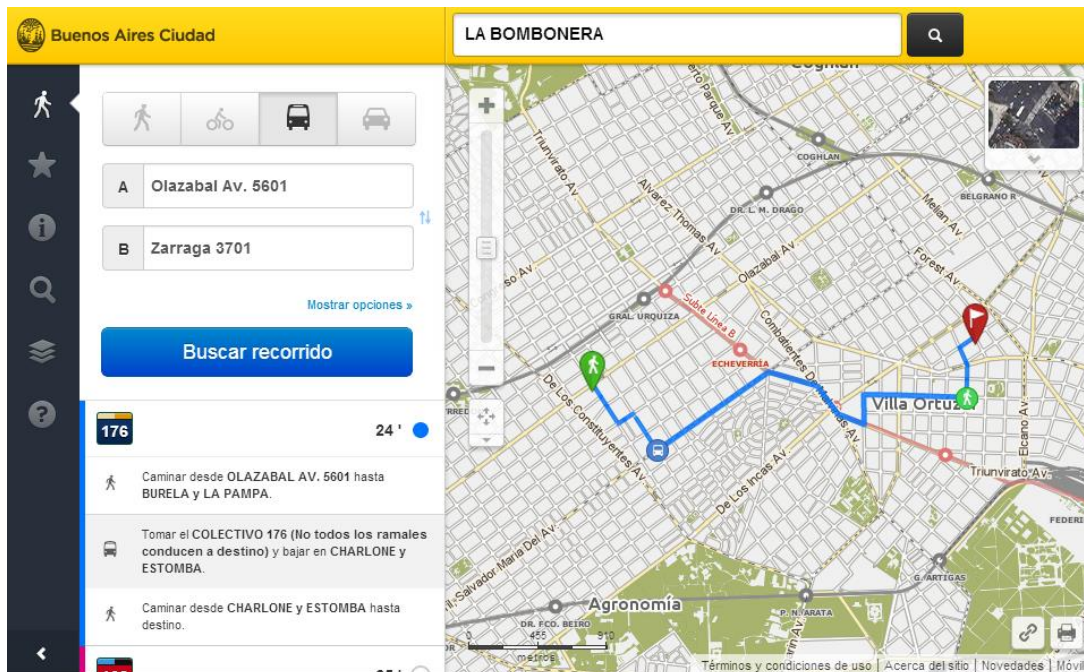


Figura 2.12: Página de busca do “Mapa Interactivo de Buenos Aires” (<http://www.mapa.buenosaires.gov.ar/>).

## 2.6 Considerações Finais

Para ilustrar as principais diferenças e funcionalidades das soluções de TIC analisadas, a Tabela 2.3 apresenta um resumo das principais características existentes em cada uma dessas ferramentas, indicando, para cada função, em quais softwares ela está presente. As funcionalidades escolhidas para comparação são: *gratuito*, indicando se a utilização do sistema implica no pagamento de licença de uso; *baseado em software livre*, indicando se o programa foi desenvolvido empregando soluções em software livre, as quais concedem ao usuário o direito de executar, copiar, distribuir, estudar, mudar e melhorar os softwares distribuídos com estas licenças; *pesquisa de linhas*, indicando se a ferramenta possibilita ao usuário consultar diretamente uma linha de ônibus ou de outro modal informando apenas o código, nome, endereço ou parada associada à linha; *pesquisa textual de rotas*, indicando se é possível consultar os possíveis caminhos existentes entre dois endereços distintos informados através de descrição textual; *pesquisa gráfica (mapa) de rotas*, indicando se é possível consultar os possíveis caminhos existentes entre dois endereços distintos informados através de marcação direta no mapa; *integrações (baldeações)*, indicando se a consulta de rotas entre dois pontos distintos considera, para a definição dos caminhos possíveis, a utilização de múltiplos modais de transporte ou os percursos a serem percorridos a pé; *múltiplos modais de transporte*, indicando se é utilizado mais de um modal de transporte (ônibus, trem, metrô ou outros) para a definição das rotas; *hora de partida e chegada*, indicando se a ferramenta apresenta os possíveis horários de partida e chegada das rotas encontradas; *tempo de viagem*, indicando se o tempo previsto de cada rota é informado; *versão web*, indicando se a ferramenta está disponível para ser utilizada diretamente pela web, acessível através de uma URL informada em um navegador; *versão web responsiva*, indicando, para aquelas aplicações que possuem versão web, se são responsivas, ou seja, apresentam as informações de forma acessível e confortável

para diferentes meios de acesso como, por exemplo, telas de diferentes tamanhos; *versão para dispositivos móveis*, indicando se a aplicação está disponível para instalação em plataformas móveis, tais como Android e iOS.

Tabela 2.3: Comparativo entre as principais funcionalidades das soluções de TIC analisadas

	<i>Poa Bus</i>	<i>Poa Transporte</i>	<i>Google Transit</i>	<i>Moovit</i>	<i>Nosso Bus</i>	<i>Antares</i>	<i>Moobly</i>	<i>Mapa Interativo de Buenos Aires</i>
Gratuito	✓	✓	✓	✓	✓	✓	✓	✓
Baseado em Software Livre	✗	✗	✗	✗	✗	✓	✗	✓
Pesquisa de linhas	✓	✓	✗	✗	✗	✓	✓	✗
Pesquisa textual de rotas	✓	✗	✓	✓	✓	✓	✓	✓
Pesquisa gráfica (mapa) de rotas	✓	✗	✓	✓	✗	✓	✓	✓
Integrações (baldeações)	✗	✗	✓	✓	✗	✓	✗	✓
Múltiplos modais de transporte	✗	✓	✓	✓	✗	✗	✗	✓
Hora de partida e chegada	✗	✗	✓	✓	✗	✓	✗	✗
Tempo de viagem	✗	✗	✓	✓	✗	✓	✗	✓
Versão web	✓	✓	✓	✓	✗	✓	✗	✓
Versão web responsiva	✗	✗	✓	✗	✗	✗	✗	✗
Versão para dispositivos móveis	✗	✗	✓	✓	✓	✗	✓	✓

Cada uma das soluções de mobilidade urbana analisadas possuem suas vantagens e desvantagens. Algumas não calculam integrações na identificação dos trajetos, outras não possuem versões disponíveis para dispositivos móveis, a grande maioria é baseada em mapas e soluções comerciais, as quais não permitem à população conhecer melhor e propor melhorias para a ferramenta que está sendo utilizada. O ideal seria que cada cidade, a exemplo de Buenos Aires, tivesse todas suas informações públicas disponíveis para a população de forma transparente e de fácil acesso, mas, infelizmente, essa não é a realidade da grande maioria das cidades de todo o mundo, especialmente cidades em que a interação do governo com a população é reduzida.

A fim de trazer à população de Porto Alegre uma nova ferramenta baseada em software livre para facilitar a vida de quem utiliza o transporte público como meio de locomoção e mesmo daqueles que não o utilizam, este trabalho propõe uma nova solução, tanto para uso em navegadores web com acesso à Internet quanto através de dispositivos móveis, a qual será apresentada no próximo capítulo.

## 3 ESPECIFICAÇÃO E PROJETO

Este capítulo apresenta a modelagem do sistema proposto neste trabalho, o qual se diferencia das demais soluções existentes por agregar, em uma mesma solução, as seguintes características principais: ser baseado em software livre; ser acessado através de qualquer navegador web ou dispositivo móvel; ser facilmente portado para qualquer cidade; e por prover informações de trajetos a pé e com integrações (baldeações).

A especificação, a qual define o que o sistema deve fazer, e quais suas restrições de desenvolvimento, é feita através dos requisitos funcionais e não funcionais da aplicação, do diagrama de casos de uso gerado a partir dos requisitos funcionais e da prototipação das telas. O projeto que, segundo Pressman (2010), cria uma representação, ou modelo do software, a partir dos requisitos, é fornecido através da arquitetura da aplicação.

Por mais simples que seja, todo e qualquer sistema deve ser modelado antes do início de sua implementação. É através dos artefatos gerados na especificação e no projeto do software que as reais necessidades são compreendidas e documentadas, proporcionando um entendimento comum de todos os envolvidos no projeto sobre o que deve ser feito, e que as informações necessárias à construção do software são fornecidas ao desenvolvedor. Além disso, é através dessa documentação que o sistema poderá ser mantido com facilidade, rapidez e de maneira correta.

### 3.1 Requisitos

Os requisitos são definidos por Sommerville e Sawyer (1997) como uma especificação do que deve ser implementado, descrevendo: como o sistema deve se comportar; quais são suas propriedades e atributos; e quais são as restrições no processo de desenvolvimento do sistema.

De acordo com Filho (2009), os requisitos podem ser classificados em: *funcionais*, que representam um comportamento que um programa ou sistema deve apresentar diante de certas ações de seus usuários; e *não funcionais*, que quantificam determinados aspectos do comportamento. Com base nessa classificação, a qual tem a intenção de encontrar, documentar e organizar o mapeamento das informações, os requisitos do sistema são definidos a seguir.



### 3.1.1 Requisitos Funcionais

Os requisitos funcionais especificam as funcionalidades do software que o desenvolvedor deve construir no sistema para permitir que usuários executem suas tarefas, satisfazendo os requisitos de negócio (WIEGERS, 2003). Eles documentam como o sistema deve reagir a entradas específicas, como deve se comportar em determinadas situações e o que o sistema não deve fazer.

Para iniciar, a principal funcionalidade disponível é possibilitar aos usuários de transporte público de Porto Alegre consultar os possíveis trajetos de ônibus entre dois pontos distintos dentro da cidade.

A interface deve possuir um conjunto de atributos para preenchimento pelo usuário, dos quais alguns são obrigatórios e outros opcionais. Para a consulta ser realizada o usuário deve informar, no mínimo, os endereços de origem e destino para que a ferramenta possa identificar os trajetos possíveis entre esses pontos. Caso algum desses parâmetros obrigatórios não esteja preenchido, então o sistema deverá indicar ao usuário a falta de preenchimento do campo específico a fim de que possa ser identificada qual informação não foi fornecida e corrigi-la.

Os endereços de origem e destino podem ser informados de duas maneiras: textual ou gráfica. No modo textual, o usuário deve preencher os campos do tipo texto com o endereço completo de cada um dos pontos para que o sistema possa, através dessa informação, identificar as coordenadas (latitude e longitude) do endereço fornecido. Para o preenchimento desses campos, à medida que o usuário digita o endereço desejado, o sistema deve autocompletar o texto digitado, exibindo os possíveis locais, a fim de permitir ao usuário selecionar uma dessas opções, facilitando, dessa forma, o preenchimento e a exatidão na escolha do endereço.

Na informação em modo gráfico dos endereços de origem e destino, o usuário deve marcar em um mapa os pontos desejados. Para facilitar a seleção, deve ser possível aumentar ou diminuir o zoom, ou mesmo modificar a área visível do mapa, a fim de auxiliar a acurácia na marcação dos pontos.

Após o usuário completar o preenchimento dos endereços de origem e destino, seja no modo textual ou gráfico, então o sistema deve, automaticamente, adequar o zoom e a área visível do mapa para que os pontos marcados fiquem visíveis para o usuário. Sempre que um endereço for preenchido em um dos modos, o sistema deverá, automaticamente, identificar esse preenchimento e incluí-lo no outro modo. Portanto, se um dos pontos for definido pelo usuário em modo textual, então o sistema deverá, automaticamente, marcá-lo no modo gráfico e vice-versa.

Além da definição dos pontos de origem e destino, o usuário também pode refinar a pesquisa através de outras opções de consulta a fim de adequar a busca do trajeto de acordo com sua necessidade.

Como critérios para refinar a pesquisa pode ser possível definir a data e a hora de partida ou de chegada, observando que se os atributos de partida forem preenchidos, então os atributos de chegada não devem ser informados, e vice-versa. Caso a data e hora de partida, ou chegada, não sejam informadas, então o sistema deverá assumir a data e a hora atual como os atributos de partida.

Outro atributo opcional é o tipo de busca de trajetos que será feito, podendo levar em conta um menor número de transferências entre ônibus ou os trajetos mais curtos.

Caso nenhuma opção esteja definida, o sistema deverá assumir um tipo de busca considerando o menor número de transferências.

O último atributo opcional é a distância máxima a qual o usuário se sujeita a caminhar para pegar o ônibus, se deslocar entre paradas e para ir até seu destino final. Os valores possíveis para este atributo devem ser: 500 m, 1 km, 2 km, 3 km e 4 km. O valor padrão, caso nenhuma opção seja definida, é 500 m.

Após a definição dos atributos obrigatórios e opcionais, o usuário deve solicitar ao sistema a realização da busca pelos trajetos possíveis de acordo com as definições realizadas. Assim que a busca for concluída, as opções de trajeto encontradas devem ser listadas para que o usuário consiga identificá-las facilmente e possa visualizar os detalhes de cada uma.

Para cada uma das opções exibidas na listagem, o sistema deve apresentar a hora de início e fim da viagem, o tempo total de duração e a identificação de cada um dos trechos percorridos, sejam eles a pé ou de ônibus, na ordem em que eles ocorrem no percurso.

Para cada um dos trajetos encontrados, o usuário poderá consultar informações detalhadas. Ao selecionar na listagem o trajeto desejado, o sistema deverá exibir os locais de início e término do percurso, informações mais detalhadas sobre cada um dos trechos, seja ele a pé ou de ônibus, e a visualização do trajeto completo em um mapa.

Os detalhes do trecho a pé devem incluir a distância total do trecho (em metros), o tempo aproximado para concluí-lo, considerando uma velocidade média de 1,341 m/s (TRANSFETY, 1997), e, para cada rua percorrida, deverá ser indicado em qual direção seguir, para que lado virar e a distância a percorrer até chegar à próxima rua ou ao final do trecho.

Os detalhes do trecho de ônibus devem incluir o código e o nome da linha, o nome da empresa responsável por ela, a data e hora em que o usuário pegará e descerá do ônibus e o tempo aproximado de viagem.

A visualização do trajeto completo no mapa, desde o endereço de origem até o endereço de destino, deverá deixar claro quais são os pontos de início e fim da viagem e quais são os trechos pertencentes ao trajeto, exibindo, para cada uma deles, quais são os pontos de início e fim.

O usuário também poderá, a qualquer momento, reiniciar a busca, fazendo com que todos os atributos preenchidos e os trajetos encontrados sejam apagados.

A ferramenta deverá estar disponível nos idiomas Inglês e Português. O idioma padrão é o Português, porém o usuário poderá, a qualquer momento, alterar o idioma selecionado.

A aplicação deve disponibilizar ao usuário a possibilidade de consultar algumas informações sobre o sistema, que são: objetivo principal; plataformas em que ela está disponível; equipe responsável pelo desenvolvimento; e agradecimentos às pessoas que contribuíram para a realização desse projeto.

Através da aplicação deve ser possível acessar, através de um *link*, um formulário para avaliar, do ponto de vista do usuário, o perfil e a usabilidade do sistema proposto.

### 3.1.2 Requisitos não funcionais

Os requisitos não funcionais incluem objetivos de desempenho e descrições de atributos de qualidade, que aumentam a descrição da funcionalidade do produto, descrevendo suas características em várias dimensões, que são importantes tanto para os usuários quanto para os desenvolvedores. Estas características incluem usabilidade, portabilidade, integridade, eficiência e robustez. Esses requisitos também descrevem interfaces externas entre o sistema e o mundo exterior, e restrições de projeto e implementação (WIEGERS, 2003).

O sistema deve ser desenvolvido empregando, preferencialmente, soluções em software livre, as quais concedem ao usuário o direito de executar, copiar, distribuir, estudar, mudar e melhorar os softwares distribuídos sob estas licenças.

O sistema deve utilizar, quando possível, *frameworks* que auxiliem na padronização de projeto, reuso e coesão, de modo a aumentar sua segurança e confiabilidade. Esses *frameworks*, caso estejam em execução nos servidores com a intenção de prover recursos para os clientes, devem, preferencialmente, ser disponibilizados na Internet através de serviços web (*webservices*).

O sistema deve ser operacionalizado através de um ou mais servidores contendo uma das inúmeras distribuições do sistema operacional GNU/Linux existentes, visto que elas oferecem maior segurança, maior confiabilidade e menor custo.

O sistema deve ser capaz de executar nos principais navegadores web da atualidade, ser acessível através de uma URL, e também através de aplicativos para dispositivos móveis.

A versão web deve ser responsiva, proporcionando ao usuário uma boa apresentação das informações, fácil leitura e navegação, com um mínimo de redimensionamento para diferentes meios de acesso, tais como, computadores pessoais, tablets e telefones celulares.

A versão para dispositivos móveis deve estar disponível, ao menos, para a plataforma Android, a qual deve ser obtida (*download*) pelo usuário através da interface web.

O sistema deve possuir baixo grau de aprendizagem, ser amigável e ser intuitivo, não exigindo qualquer tipo de treinamento prévio aos usuários que irão operá-lo.

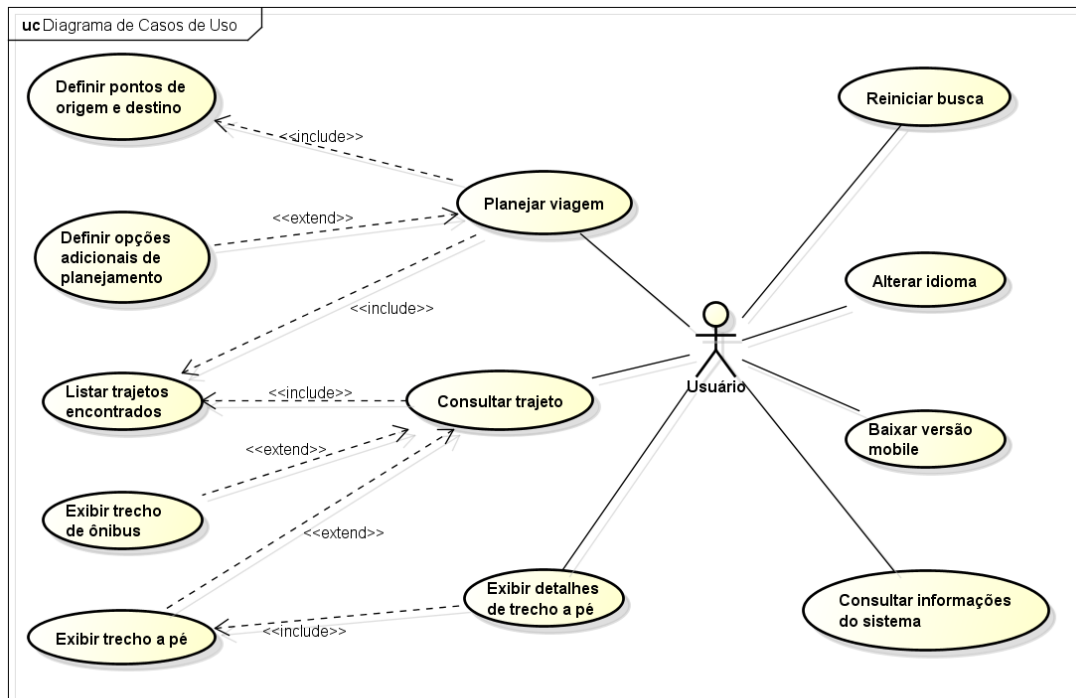
O tempo de resposta do sistema não deve ultrapassar 10 segundos, que é o limite de tempo em que é possível manter a atenção do usuário (NIELSEN, 1993).

### 3.2 Diagrama de casos de uso

A linguagem UML (*Unified Modeling Language*) tem como objetivo auxiliar engenheiros de software a definir as características do software, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado (GUEDES, 2004).

O diagrama de casos de uso é o mais geral e informal da UML, tendo como objetivo apresentar, através de uma linguagem simples e de fácil compreensão, uma visão externa geral das funções e serviços que o sistema deve oferecer aos usuário, sem se preocupar em como essas funções e serviços serão implementados, possibilitando a mesma compreensão por parte de todos os envolvidos no processo (GUEDES, 2004).

A Figura 3.1 apresenta o diagrama de casos de uso feito com base nos requisitos funcionais apresentados na seção anterior, ilustrando os tipos de usuários que interagem com o sistema (atores) e as funções que podem ser realizadas (casos de uso). A documentação dos casos de uso não foi elaborada, pois ela acabaria por repetir o que já foi descrito nos requisitos funcionais, apenas com a diferença de estar em um formato mais simples e mais específico, não agregando grande valor aos artefatos de documentação.



powered by Astah

Figura 3.1: Diagrama de Casos de Uso.

### 3.3 Prototipação de interface

A última etapa de especificação deste trabalho é a prototipação, que consiste em desenvolver rapidamente um “rascunho” do que será o sistema de informação quando este estiver finalizado. Um protótipo ilustra a interface do software a ser desenvolvida; como as informações são inseridas e recuperadas no sistema; e apresenta alguns exemplos com dados fictícios (GUEDES, 2004).

As próximas duas seções apresentam os protótipos das telas das versões web e *mobile*, as quais foram construídas com base nos requisitos funcionais e no diagrama de casos de uso através do *Pencil* (PENCIL, 2012), uma ferramenta gratuita e de código aberto destinada à prototipação de interfaces. A construção desses artefatos permite avaliar se o software a ser desenvolvido atende as necessidades especificadas.

Na maioria dos protótipos foram utilizados balões de texto como uma forma de disponibilizar informações adicionais sobre cada uma das telas. Portanto, eles são meramente informativos, não sendo visíveis na implementação da solução final.

#### 3.3.1 Interface web

A Figura 3.2 ilustra o protótipo da tela principal do sistema representada pelo caso de uso *Planejar viagem*. Nela é possível notar as principais divisões da interface: *cabeçalho*, contendo o logotipo e os *links* para as demais opções do sistema; *painel de entrada de dados*, local por onde o usuário informará os campos obrigatórios e opcionais da pesquisa, além do botão responsável por realizar a pesquisa de trajetos; *painel auxiliar*, região contendo informações adicionais para facilitar a interação do usuário com o sistema, apresentará os erros e avisos e também listará os trajetos encontrados; *mapa*, área que apresentará o mapa da cidade de Porto Alegre, permitirá a interação do usuário com ele e também apresentará o trajeto selecionado na listagem; e, por fim, o *rodapé*, contendo apenas o *link* para o formulário utilizado para avaliar o sistema.

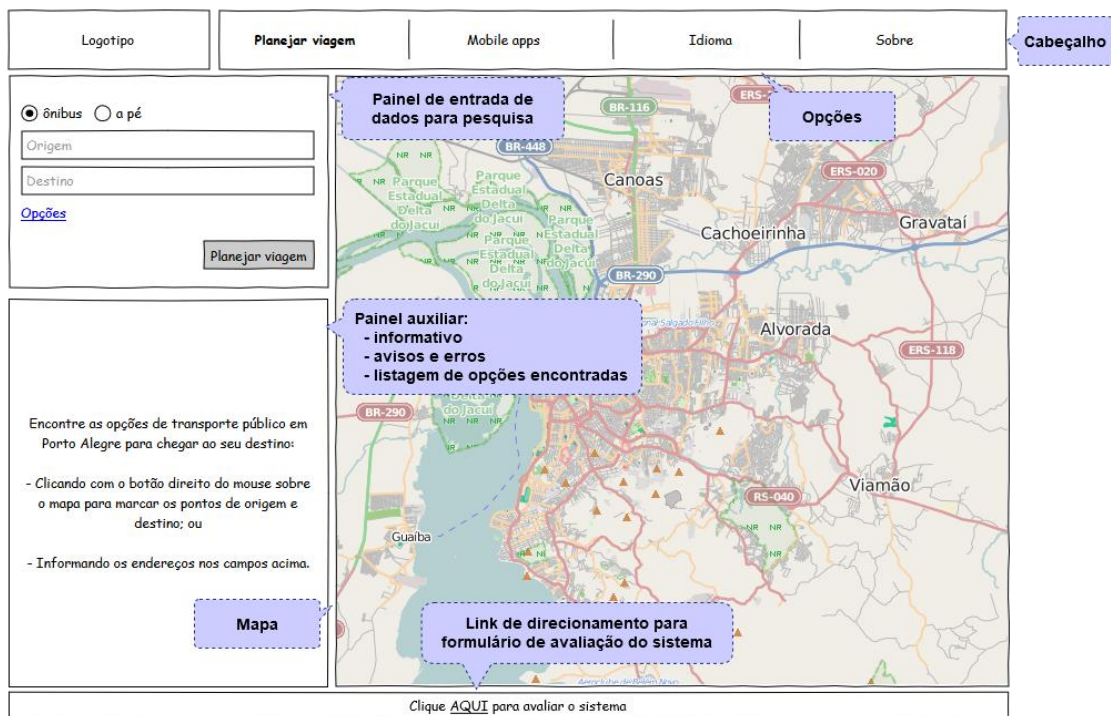


Figura 3.2: Interface web - tela de planejamento de viagens.

A Figura 3.3 ilustra como serão apresentadas as opções adicionais de pesquisa na tela. Por padrão, elas não estarão visíveis e serão apresentadas apenas quando o usuário clicar no link “Opções”, disponível no *painel de pesquisa*. Esse comportamento da interface foi adotado para torná-la mais limpa, simples e prática, visto que as opções de data e hora de partida ou chegada, o tipo de busca e o percurso máximo percorrido a pé não são obrigatórios e podem ser mantidos com os valores padrão definidos nos requisitos funcionais.

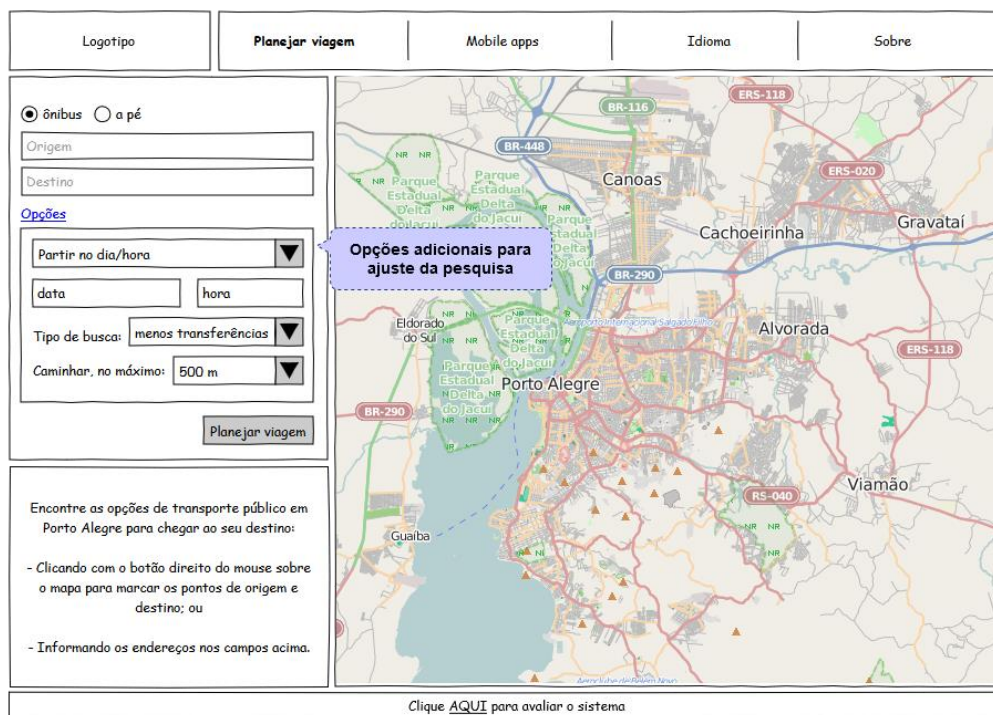


Figura 3.3: Interface web - opções adicionais de pesquisa.

A Figura 3.4 apresenta como deve ser o preenchimento da pesquisa em modo textual dos endereços de origem e destino e indicando como devem ser apresentadas as possíveis opções de endereço de acordo com o texto digitado pelo usuário.

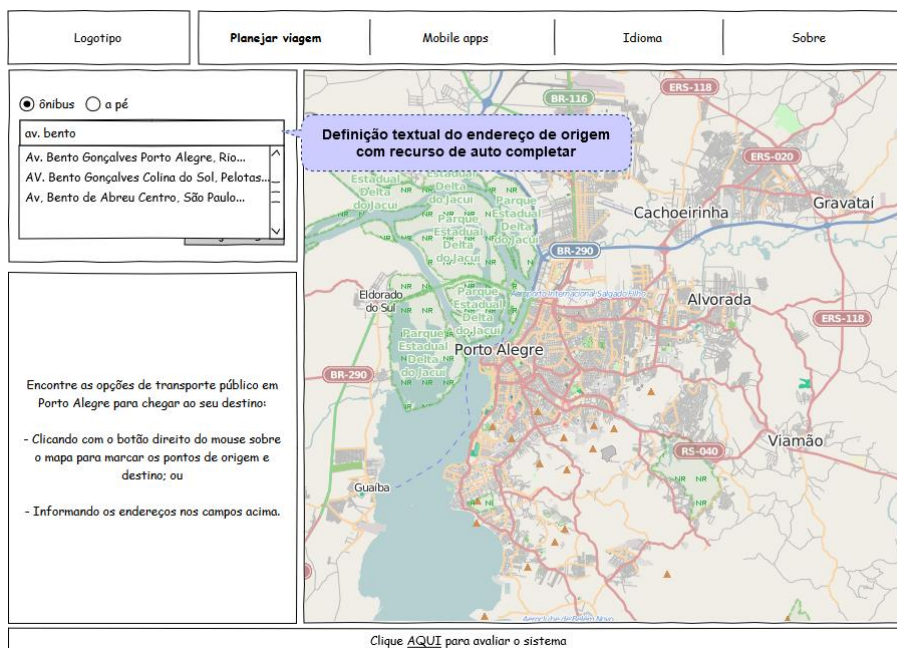


Figura 3.4: Interface web - pesquisa textual.

A Figura 3.5 apresenta o endereço de origem já preenchido, tanto no modo textual, quanto no mapa, visto que estas opções estão sincronizadas, conforme definido nos requisitos funcionais. Também ilustra onde devem ser apresentadas as informações de erro e aviso. Nesse protótipo, especificamente, está ilustrada uma mensagem de erro exibida após o usuário clicar no botão “Planejar viagem” indicando a falta de preenchimento do endereço de destino.

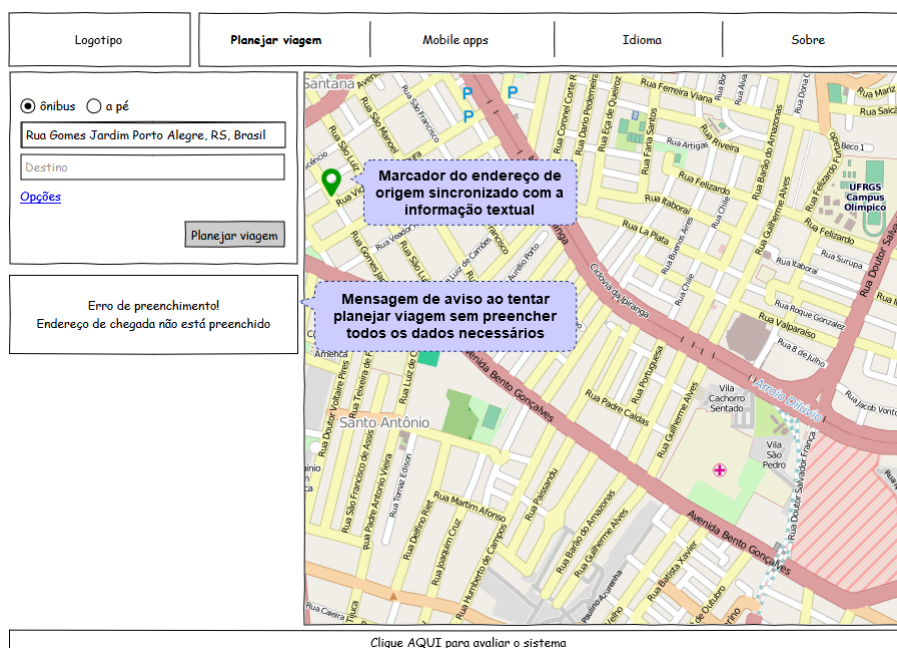


Figura 3.5: Interface web - marcação automática do endereço no mapa e exibição de mensagens de erro e aviso.

A Figura 3.6 apresenta o botão responsável por permitir a realização de uma nova pesquisa, fazendo com que os dados preenchidos pelo usuário e os trajetos exibidos sejam apagados. Esse botão estará visível apenas quando ao menos uma informação estiver sido informada pelo usuário. Enquanto isso não ocorrer, o sistema não irá exibi-lo. Nessa tela também estão sendo ilustradas algumas opções possíveis de interação com o mapa. Clicando com o botão direito do mouse, em um determinado local do mapa, deve ser possível definir o ponto de origem ou de destino e aumentar ou diminuir o zoom sobre a área desejada. Ao marcar o ponto de origem, ou destino, no mapa, o sistema deve, automaticamente, preencher o campo para informação textual. Além dessas opções, o usuário também poderá navegar no mapa mantendo o botão esquerdo do mouse pressionado sobre um determinado ponto, como se estivesse segurando o mapa, e arrastá-lo a fim de modificar a área visível. Também deve ser possível modificar os pontos de origem e destino apenas mantendo o botão do mouse pressionado sobre a marcação desejada e arrastando-a para o outro local de interesse.

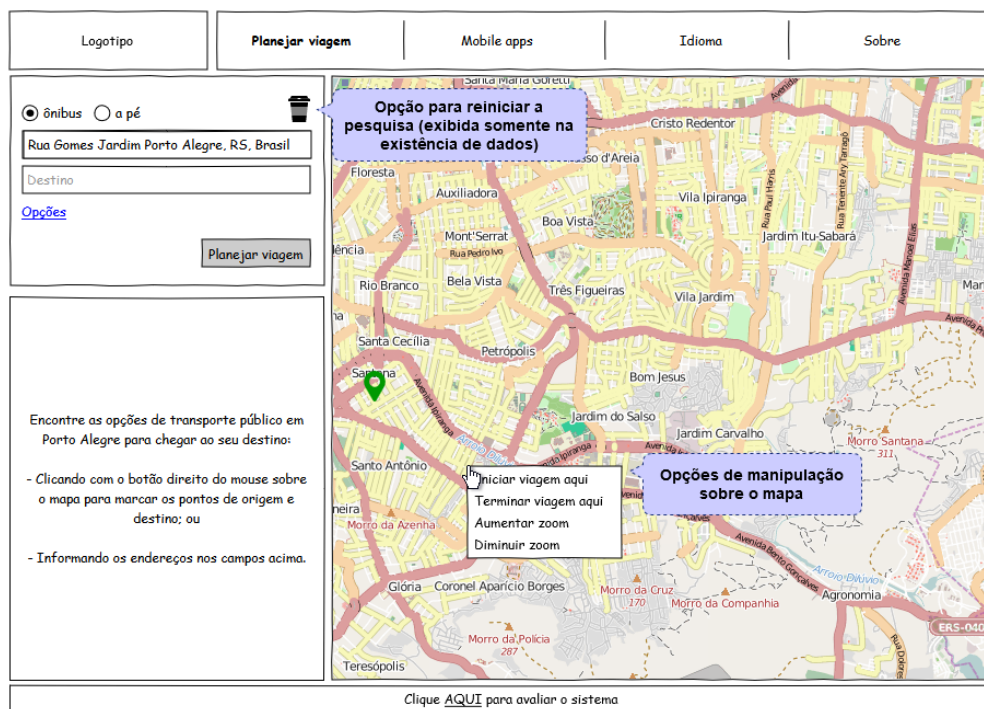


Figura 3.6: Interface web - reiniciar pesquisa e interagir com o mapa.

A Figura 3.7 apresenta os pontos de origem e destinos já definidos, seja no modo texto quanto no modo gráfico. Com a definição do segundo ponto, o sistema realiza, automaticamente, o auto ajuste do zoom e da área visível no mapa a fim de exibir os pontos de origem e destino. Nesse momento o planejamento da viagem está apto a ser efetuado.



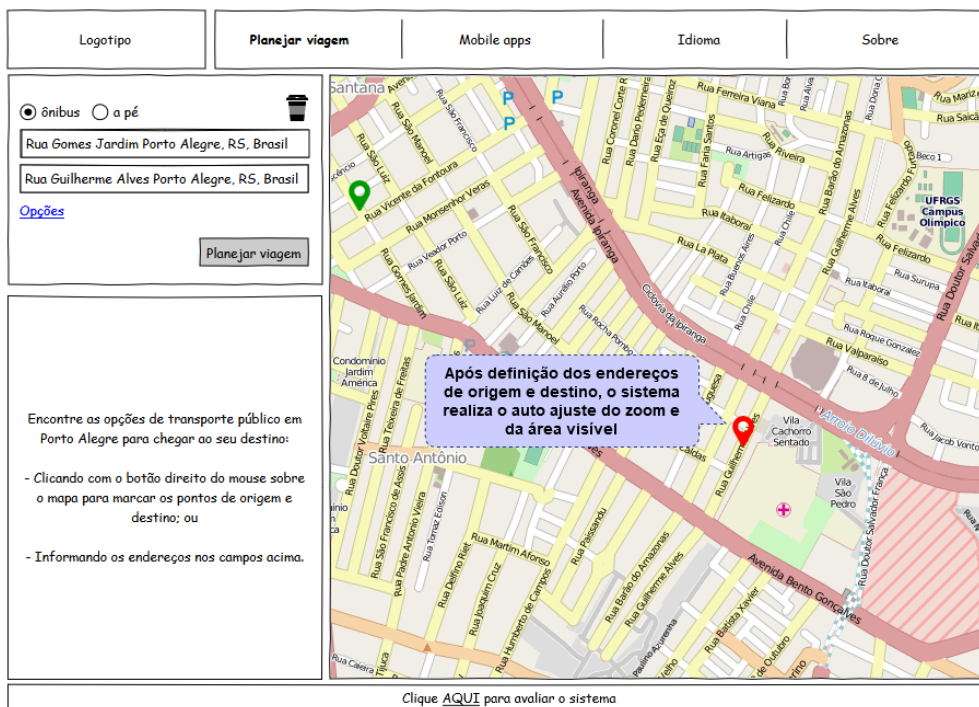


Figura 3.7: Interface web - pontos de origem e destino preenchidos.

A Figura 3.8 mostra o resultado apresentado após uma pesquisa bem sucedida. Nesse exemplo, o sistema encontrou três opções de trajeto, sendo que nas duas primeiras o usuário necessita de apenas um ônibus (linha 375 ou 347), enquanto na última é necessário dois (linhas 340 e T2). As opções de trajeto são exibidas em cores distintas para facilitar a identificação de cada uma delas.

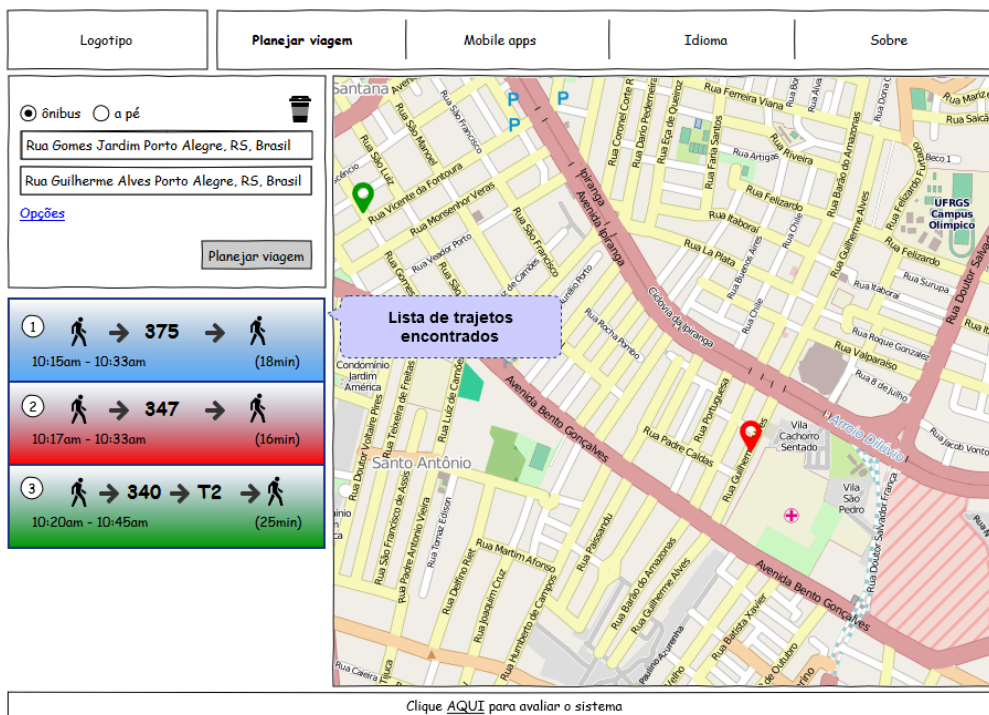


Figura 3.8: Interface web - listagem dos trajetos encontrados.

A Figura 3.9 apresenta a seleção de um dos trajetos listados. Ao fazer isso o sistema exibe, de forma textual e gráfica, os detalhes do trajeto selecionado, conforme indicado na especificação dos requisitos funcionais.

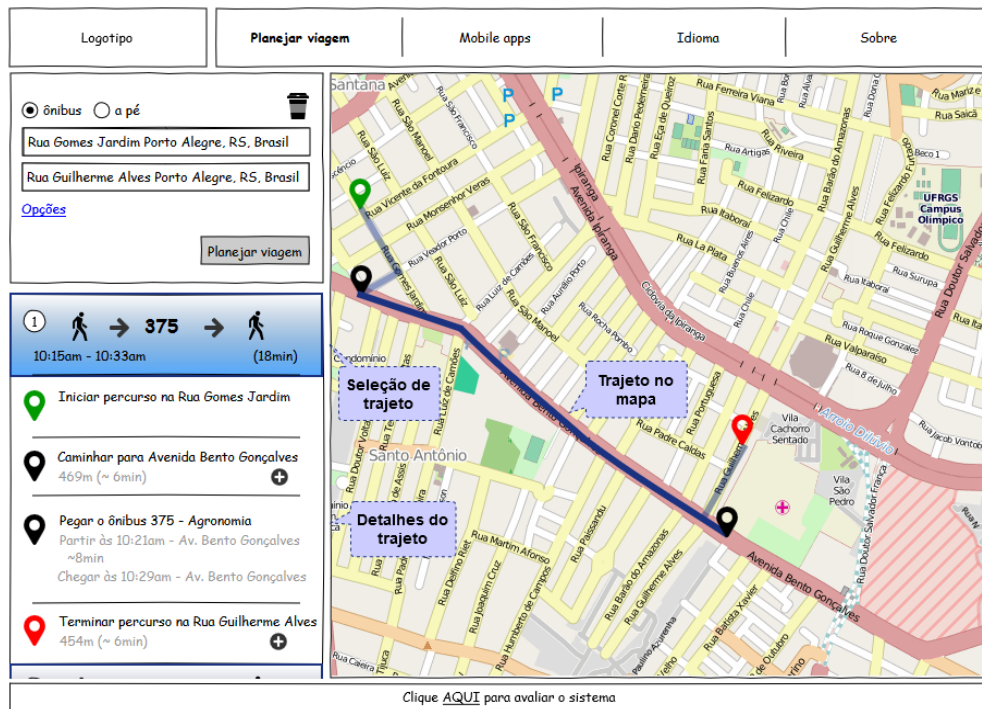


Figura 3.9: Interface web - seleção de trajeto.

A Figura 3.10 indica como deve ser exibido os detalhes do trajeto a pé. Por padrão, essas informações não estarão visíveis para tornar a tela mais limpa, mas poderão ser exibidas, caso o usuário deseje, clicando sobre o [link](#) indicado na figura. Ao clicar sobre o [link](#) novamente as informações ficarão invisíveis novamente.

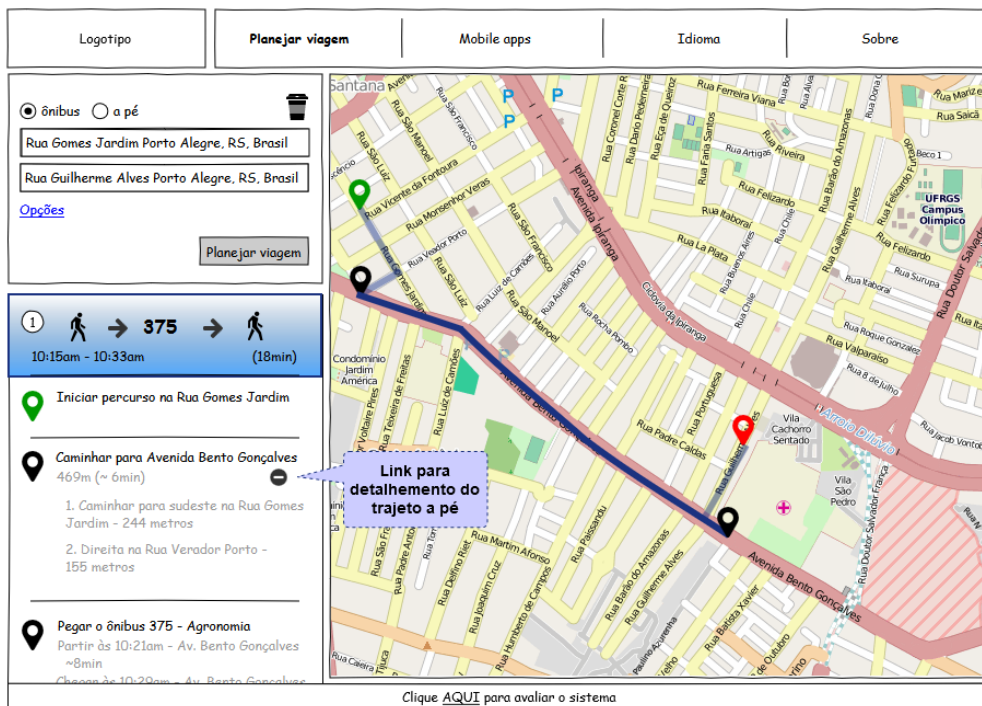


Figura 3.10: Interface web - detalhes do trajeto a pé.

A Figura 3.11 apresenta outra das quatro opções de acesso disponíveis no cabeçalho da tela. A opção *Mobile apps*, ao ser selecionada, exibe os *links* para que o usuário possa realizar o *download* das versões disponíveis para dispositivos móveis. Ao clicar sobre um deles, o usuário baixará para seu dispositivo a versão *mobile* escolhida.

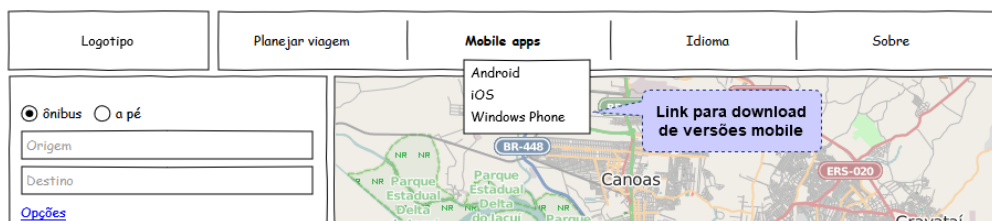


Figura 3.11: Interface web - baixar versão *mobile*.

A Figura 3.12 exibe as opções de idioma. Inicialmente, a ferramenta terá apenas opções pelos idiomas Português e Inglês, tendo o Português como o padrão.

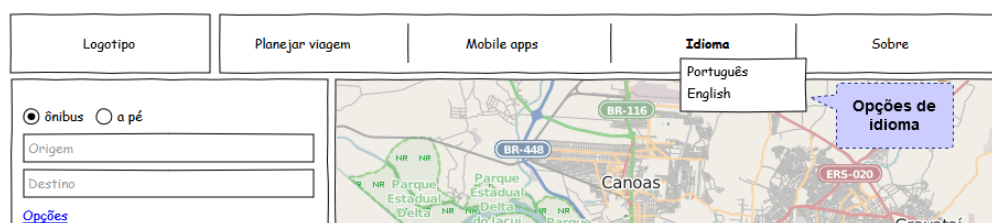


Figura 3.12: Interface web - seleção de idioma.

O último protótipo da interface web (Figura 3.13) apresenta a organização com que as informações sobre o sistema devem ser exibidas quando o usuário clicar sobre a última opção do cabeçalho e como elas podem ser facilmente acessadas através dos links disponíveis no lado esquerdo da tela.



Figura 3.13: Interface web - informações do sistema.

### 3.3.2 Interface para dispositivos móveis

Similar ao protótipo web da Figura 3.2, a Figura 3.14 ilustra como deve ser apresentada a tela principal do sistema quando acessada através de um dispositivo móvel. A diferença está em como as informações estão dispostas na tela, pois algumas funcionalidades sempre exibidas na versão web, são apresentadas somente quando o usuário solicita, visto que as dimensões de um aparelho móvel são reduzidas em relação a uma estação de trabalho.

A disposição das informações na tela se diferencia da versão web na apresentação do *cabeçalho*. Por padrão, é apresentado apenas o logotipo do sistema, não exibindo os *links* para as demais opções, apenas quando acessado através do *link* disponível no canto superior direito da tela. O *mapa* é exibido apenas quando o usuário clica nos *links* para marcação dos endereços origem e destino, ou no *link* para visualização do trajeto no mapa.

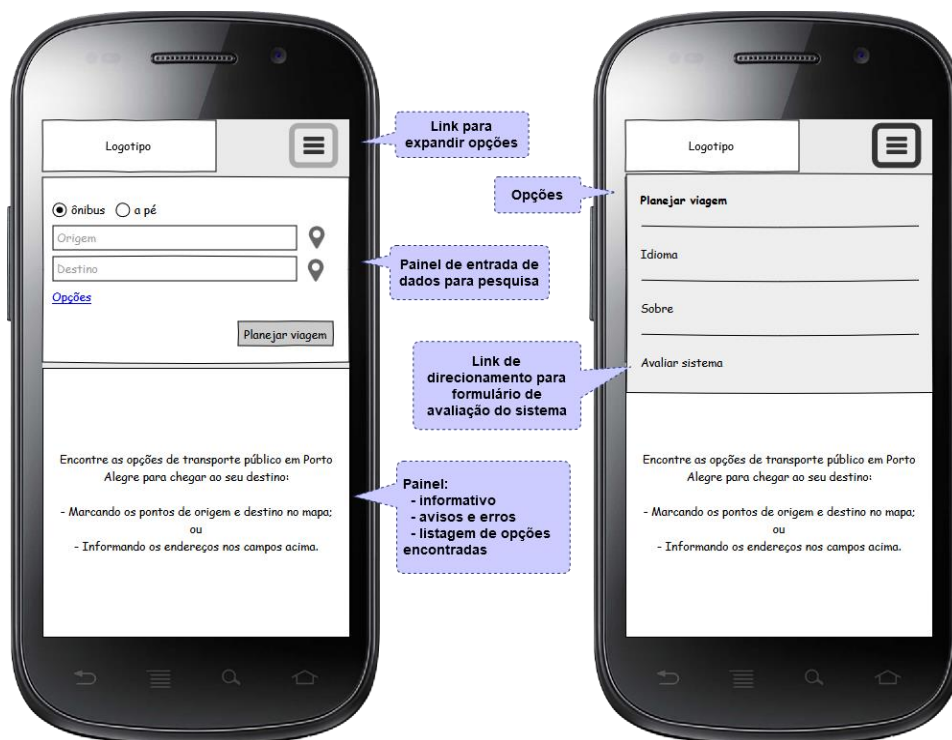


Figura 3.14: Interface *mobile* - tela de planejamento de viagens.

A Figura 3.15 ilustra a forma com que as informações adicionais de pesquisa e o preenchimento dos endereços de origem e destino devem ser apresentados ao usuário, não havendo diferenças em relação à versão web.

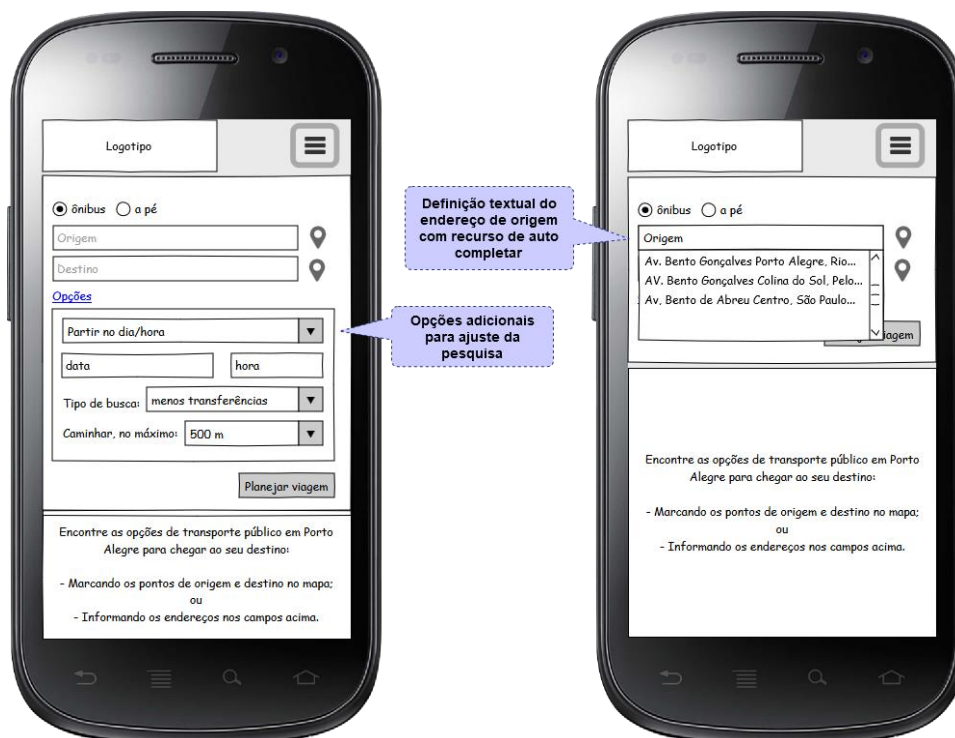


Figura 3.15: Interface *mobile* - opções adicionais de pesquisa e preenchimento textual dos endereços de origem e destino.

A Figura 3.16 apresenta como devem ser exibidas as informações de erro, as informações de aviso e o botão para reiniciar a pesquisa, que não necessitam de explicação adicional, pois são similares à versão web.

A mesma figura também retrata como o usuário deve fazer a seleção dos pontos de origem e destino no mapa, a qual possui uma apresentação diferenciada em relação à versão web. Por padrão, o mapa não é visível ao usuário, sendo apresentado apenas quando requisitado através de um dos *links* dispostos ao lado das caixas textuais para informação dos endereços. Ao clicar em um dos *links*, o sistema apresenta a tela que permite a seleção gráfica dos pontos de origem e destino no mapa, a qual será exibida sem qualquer endereço marcado, mesmo que este já esteja definido na caixa textual. Na tela para seleção gráfica dos pontos será possível retornar a tela anterior sem realizar qualquer alteração, navegar sobre o mapa e marcar um endereço de origem ou destino, fazendo com que essa nova informação também seja atualizada na caixa textual.

No mapa, o usuário pode: modificar a área visível, deslizando o dedo em direção à região desejada; aumentar ou diminuir o zoom, deslizando os dedos indicador e médio, simultaneamente, em sentidos contrários, afastando-os um do outro (aumentar zoom) ou aproximando-os (diminuir zoom); marcar o ponto, pressionando o dedo sobre o local desejado.



Figura 3.16: Interface *mobile* - reiniciar pesquisa, exibição de mensagens e interação com o mapa.

A Figura 3.17 mostra como os trajetos devem ser listados, selecionados e exibidos no mapa. A listagem e a visualização dos detalhes dos trajetos segue o mesmo modelo da interface web, porém, a visualização no mapa é diferente, não sendo apresentada por padrão. Para visualizar o mapa o usuário deve pressionar o botão “*Visualizar trajeto no mapa*”, exibido nos detalhes de cada um dos trajetos.



Figura 3.17: Interface *mobile* - listagem e seleção de trajetos.

Os detalhes do trajeto a pé e a exibição do trajeto no mapa são ilustrados na Figura 3.18. Os detalhes do trajeto a pé são apresentados apenas quando o usuário acessa o *link* para esse fim. O trajeto no mapa é exibido apenas quando o usuário pressiona o botão “*Visualizar trajeto no mapa*”, fazendo com que uma nova tela seja exibida, a qual apresenta a opção de voltar para a tela anterior e exibe o trajeto no mapa, o qual permite o ajuste da área visível e modificar o *zoom*.

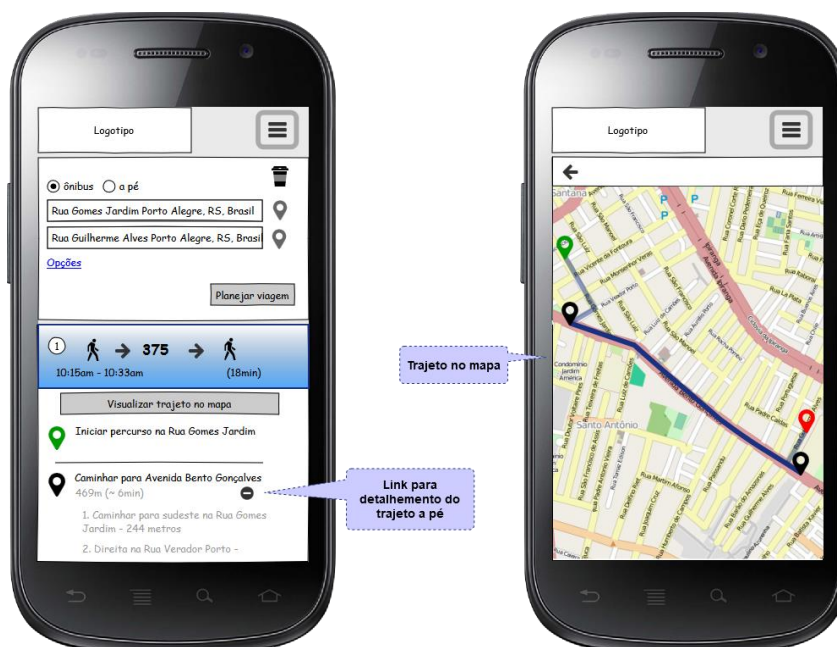


Figura 3.18: Interface *mobile* – detalhes de trajeto a pé e exibição no mapa.

O último protótipo, ilustrado na Figura 3.19, apresenta as demais opções disponíveis no cabeçalho do sistema, as quais são acessíveis através do *link* exibido no canto superior direito na tela. Através dessas opções, é possível modificar o idioma da aplicação alternando entre Inglês e Português, visualizar as informações adicionais do sistema e também acessar o formulário de avaliação.

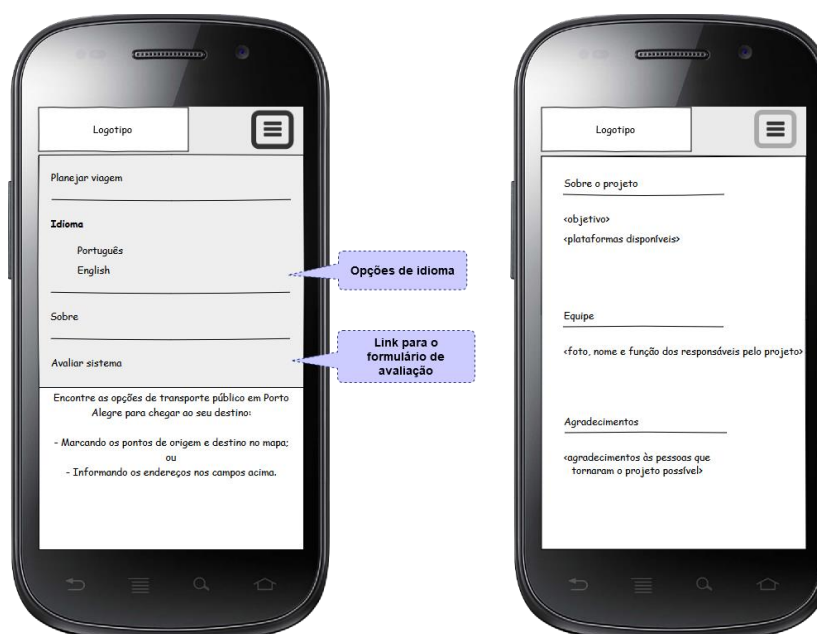


Figura 3.19: Interface *mobile* – selecionar idioma e informações do sistema.

### 3.4 Arquitetura do sistema

De acordo com Bass (2003), arquitetura de software de um programa, ou sistema computacional, é a estrutura, ou as estruturas, do sistema, na qual compreendem os componentes de software, as propriedades visíveis externamente desses componentes e as relações entre eles.

O sistema proposto neste trabalho deve abordar, no mínimo, os seguintes padrões arquiteturais: *modelo cliente-servidor* para a distribuição das cargas de trabalho; *padrão Model-View-Controller (MVC)* para separar a representação da informação da interface do usuário; e *modelo de comunicação e tipos de dados fornecidos pelas aplicações* para definir como os diferentes componentes interagem entre si. As próximas três seções fornecem maiores informações sobre cada um deles e apresentam como eles devem ser empregados no trabalho.

#### 3.4.1 Modelo cliente-servidor

Visando distribuir as tarefas e cargas de trabalho, armazenar dados de forma centralizada e possibilitar o funcionamento simultâneo do sistema em múltiplos clientes, este trabalho deve empregar uma arquitetura *cliente-servidor*. Essa arquitetura é definida por Tanenbaum (2007) como um modelo onde os processos são divididos em dois grupos: um servidor, que implementa um ou mais serviços específicos; e um cliente, que requisita um serviço ao servidor enviando-lhe uma requisição e, na sequência, esperando por uma resposta. A Figura 3.20 mostra uma representação de uma comunicação entre clientes e servidores.

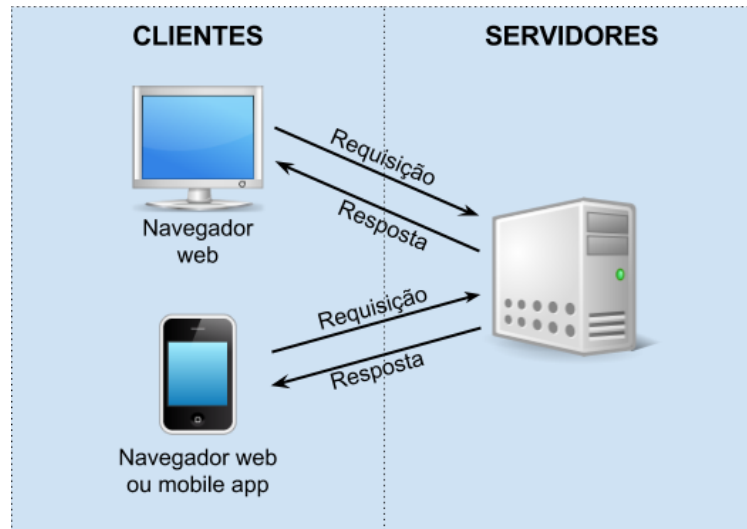


Figura 3.20: Comunicação cliente-servidor.

Existem oito módulos principais que compõem este trabalho, sendo seis deles no lado servidor e dois deles no lado cliente. Os seis módulos do lado servidor são responsáveis pela disponibilização de serviços essenciais para o fornecimento da solução completa, a qual será consumida pelos clientes. O lado servidor da aplicação deve ser composto por: (i) um servidor responsável pelo fornecimento de dados cartográficos; (ii) uma API para autocompletar endereços; (iii) uma API de conversão de endereços; (iv) uma API de imagens cartográficas; (v) uma API para o cálculo de rotas; e (vi) um servidor web. O lado cliente é composto por: (i) navegadores web; e (ii) aplicações *mobile*.



A comunicação dos servidores com os clientes deve ser realizada através da Internet por meio dos protocolos de comunicação TCP/IP. Já a forma de comunicação entre os servidores pode se dar de duas formas: através da Internet (TCP/IP); ou, caso não seja possível devido a restrições de implementação, a troca de informações deve ser feita manualmente pelo administrador do sistema através da exportação e importação de arquivos em formatos compatíveis entre os módulos.

O servidor cartográfico tem a finalidade de fornecer as informações geoespaciais do planeta, tais como prédios, casas, ruas e rodovias, para as APIs que compõem a solução. Essas APIs devem manter a consistência das informações entre elas, não podendo haver divergências entre as informações que são apresentadas.

A API para autocompletar endereços tem a função de receber um determinado texto e retorna previsões de locais. Para cada solicitação recebida, a API envia a requisição ao servidor de dados cartográficos, recebe uma resposta e envia-a de volta ao cliente para que este possa exibi-la ao usuário.

A API de conversão de endereços é responsável por receber um dado endereço na forma textual e retornar suas coordenadas geográficas (latitude e longitude) e vice-versa. A cada solicitação de endereço pelo cliente a API envia uma requisição ao servidor de dados cartográficos, recebe uma resposta dessa requisição e, em seguida, devolve a resposta para o cliente.

A API de imagens cartográficas deve fornecer os dados cartográficos em forma de imagens para que possam ser exibidas na tela do cliente. A geração dessas imagens não deve ser feita a cada requisição do cliente, mas sim periodicamente, pois o tempo necessário para gerá-las tornaria a solução inviável. Para criá-las, o servidor que está hospedando esta API deve requisitar ao servidor de dados cartográficos as informações do mapa, receber a resposta com os dados desejados e, em seguida, gerar as imagens a partir desses dados, armazená-las localmente e disponibilizá-las para os clientes.

A última API, cálculo de rotas, a compor a solução é responsável por definir a rota a ser percorrida pelo usuário, seja a pé ou de ônibus, e disponibilizar os trajetos encontrados aos clientes. Para fornecer esse serviço, o servidor em que a API está localizada deve ser alimentado com os dados cartográficos, seja através de requisição e resposta feita pela Internet, ou mesmo pelo fornecimento manual de arquivos geoespaciais em formato conhecido. Também deve ser capaz de receber os dados e informações geográficas que descrevam os transportes públicos da região desejada para poder alimentar sua base de dados local e conseguir realizar os cálculos.

Por fim, deve haver um servidor web responsável por receber requisições de páginas HTML e enviá-las como resposta aos navegadores web. O navegador então exibe a página HTML na tela para que o usuário possa interagir com ele e, dessa forma, possa, indiretamente, acessar as APIs responsáveis por complementar o funcionamento da aplicação.

Os clientes, sejam eles navegadores web, ou aplicações *mobile*, irão consumir as APIs para que a aplicação possa realizar todas as funcionalidades que se propõe. Os navegadores também irão se comunicar com o servidor web para receber as páginas necessárias para visualização e interação com o sistema.

A Figura 3.21 apresenta uma possível estrutura física para o sistema proposto, onde as APIs para autocompletar endereços, converter endereços e fornecer imagens cartográficas estão sendo disponibilizadas em uma única máquina, enquanto os demais

módulos possuem uma máquina para cada um deles. A estrutura física apresentada na figura não implica que a configuração final necessite ser exatamente esta. É possível ter múltiplos serviços sendo executados em uma única máquina, ou mesmo múltiplas máquinas executando um único serviço.

As linhas contínuas com setas bidirecionais representam a comunicação existente entre os servidores e dos clientes, sejam eles navegadores web ou aplicações *mobile*, com os módulos em execução no lado servidor. A linha contínua com seta unidirecional representa a importação do arquivo de informações sobre transporte público na API para o cálculo de rotas. E as linhas tracejadas representam uma possibilidade, ou seja, a API para o cálculo de rotas pode receber as informações cartográficas diretamente do servidor ou através de um arquivo texto gerado previamente.

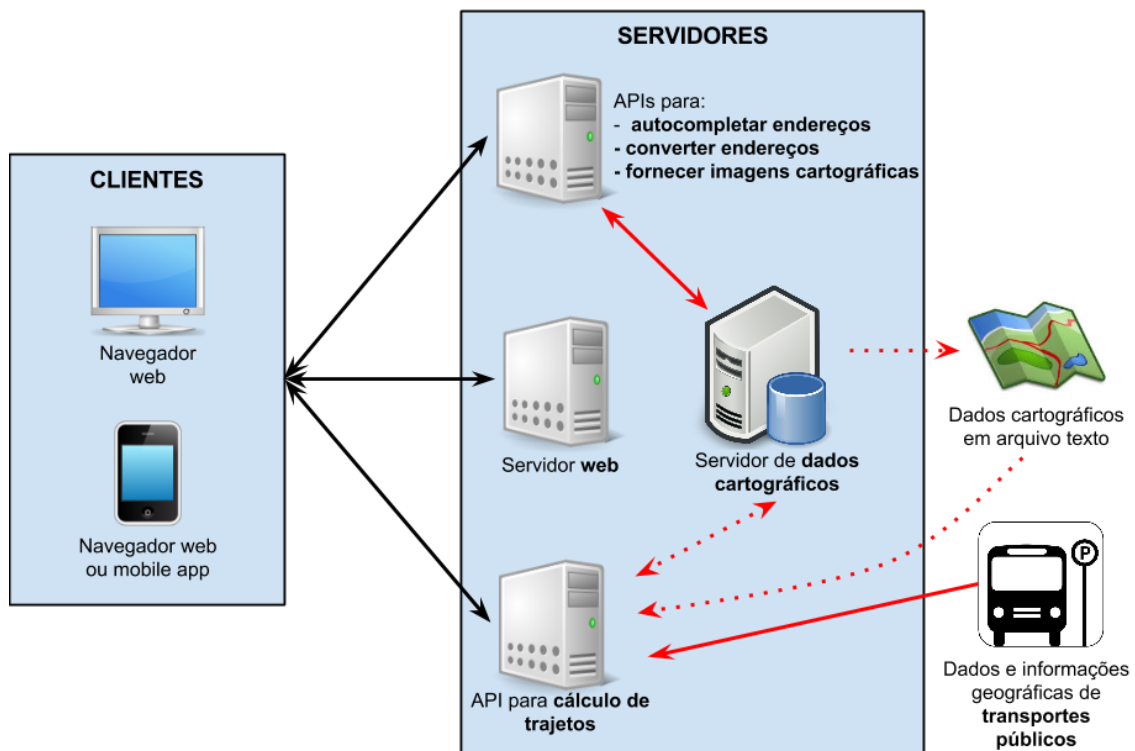


Figura 3.21: Modelo cliente-servidor deste trabalho.

### 3.4.2 Padrão *Model-View-Controller* (MVC)

O padrão MVC é um modelo de arquitetura de software que separa a interface do usuário da representação da informação, estruturando o sistema em três camadas distintas. O *modelo* possui o conteúdo e a lógica de negócio da aplicação. A *visão* contém todas as funções de interface e é responsável por apresentar o conteúdo ao usuário. E, por fim, a camada de *controle*, que gerencia o acesso às camadas de modelo e visão, coordenando o fluxo de dados entre elas (PRESSMAN, 2010).

Esse modelo é empregado nos módulos de conversão de endereços, cálculo de trajetos e também no servidor web. Nesses módulos as três camadas inteiras estão no servidor, onde o controlador recebe uma requisição feita pelo usuário, o modelo trata da lógica de negócio para retornar o conteúdo requerido e, por fim, a visão retorna a saída requisitada no formato desejado pelo usuário.

### 3.4.3 Modelo de comunicação e tipos de dados fornecidos pelas aplicações

Todos os módulos da aplicação que interagem com os clientes (API para autocompletar endereços, API para conversão de endereços, API de imagens cartográficas, API para cálculo de trajetos e servidor web) devem empregar o protocolo de comunicação HTTP, o qual é a base para comunicação de dados na Web.

O servidor web e a API de imagens cartográficas devem receber uma mensagem de requisição HTTP e retornar, no corpo da mensagem de resposta, os conteúdos de hipermídia necessários para apresentar o sistema ao usuário. No caso do servidor web, o conteúdo de hipermídia retornado será, em grande parte, páginas HTML que serão recebidas pelos navegadores web. Já a API retornará exclusivamente imagens, que serão recebidas tanto pelos navegadores web quanto pelas aplicações *mobile* para serem apresentadas nas telas dos clientes.

As APIs para autocompletar endereços, para conversão de endereços e para cálculo de trajetos devem ser disponibilizadas aos clientes na forma de um serviço web RESTful, o qual deve receber requisições HTTP e responder às requisições no formato JSON.

## 3.5 Considerações finais

Neste capítulo foi apresentada a especificação do sistema a ser desenvolvido usando como técnicas a especificação de requisitos, o diagrama de casos de uso e a prototipação de interfaces. Além disso, também foi fornecida a arquitetura do sistema, que está baseada no modelo cliente-servidor.

O próximo capítulo descreve a implementação do sistema conforme a especificação discutida. Como mencionado nos requisitos funcionais, procurar-se-á, na medida do possível, empregar ferramentas e softwares livres já existentes e bem estabelecidos para a realização de tarefas de conversão de endereços, armazenamento e distribuição de dados cartográficos e cálculo de rotas.

## 4 IMPLEMENTAÇÃO

Com base nas informações de especificação e projeto, este capítulo traz os detalhes da solução proposta neste trabalho, onde são apresentadas: a arquitetura completa da solução, descrevendo as tecnologias empregadas em cada um dos módulos envolvidos e o modelo físico da solução final; os detalhes da implementação no lado servidor, mostrando as instalações e configurações feitas para fornecer os módulos necessários para o funcionamento do sistema; os detalhes da implementação dos aplicativos web e *mobile*; e, por fim, as considerações finais encerrando o capítulo e trazendo informações adicionais sobre a implementação da solução.

### 4.1 Arquitetura completa da solução

Sabendo-se o que, e como, a solução deve ser construída, o próximo passo é definir quais ferramentas serão utilizadas para implantar o sistema e como elas interagirão entre si. Uma definição de requisito não funcional que tem papel fundamental na escolha dessas ferramentas é que devem ser empregadas, preferencialmente, soluções em software livre.

Com base no que foi especificado e projetado, a primeira escolha para a implantação da solução é o *OpenStreetMap* (OSM), que é uma ferramenta colaborativa destinada a criar e distribuir gratuitamente dados geográficos do mundo inteiro (seção 2.3). O *servidor OSM*, o qual possui os dados cartográficos armazenados em sua base de dados, será o responsável por disponibilizar arquivos no formato OSM XML, os quais serão gerados através da ferramenta *Overpass API*, para alimentar a ferramenta de busca por trajetos. O OSM também será responsável por disponibilizar os dados para a ferramenta *Mapnik* gerar imagens (*tiles*) do mapa cartográfico com diferentes níveis de zoom em *servidores de tile*. Essas imagens serão utilizadas pelos clientes web e *mobile* para exibir o mapa, permitindo ao usuário navegar sobre ele, modificando o zoom e alterando a área visível, para selecionar pontos de origem e destino e também visualizar os trajetos encontrados.

A próxima ferramenta escolhida para a construção do trabalho é o *OpenTripPlanner* (OTP), que é um projeto de código aberto utilizado para planejar viagens multimodais e para analisar redes de transporte (seção 2.4). Este trabalho utiliza três dos quatro módulos principais da ferramenta: *GraphBuilder*, *Routing* e *OTP Routing API*.

A ferramenta escolhida para disponibilizar a aplicação web para acesso dos usuários através de navegadores, a qual será retratada na seção 4.3.1, foi o *servidor web Apache*.

As últimas tecnologias necessárias no lado servidor são os serviços de autocompletar endereços e de geocodificação, utilizados para o preenchimento

automático de endereços em campos texto e para converter nomes e endereços em coordenadas, respectivamente. A solução existente em software livre para atender a esses requisitos é o *Nominatim* (seção 2.3). Contudo, essa ferramenta ainda não está madura o bastante para fornecer uma solução robusta. Sendo assim, a ferramenta escolhida foi o *serviço de pesquisa de locais e de geocodificação do Google Maps*, o qual é disponibilizado pela própria Google.

Com base na Figura 3.21, onde foi apresentada a arquitetura cliente-servidor deste trabalho, e no que foi descrito nesta seção, a Figura 4.1 ilustra a arquitetura da solução com as tecnologias empregadas nos servidores e destaca a comunicação entre eles.

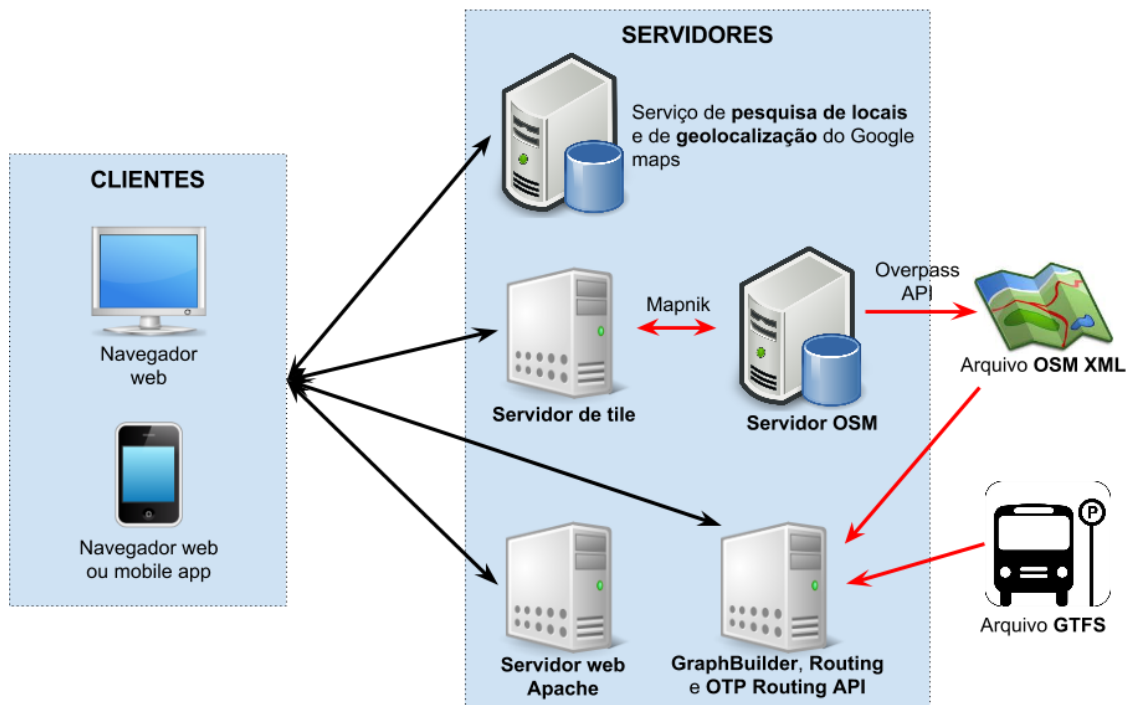


Figura 4.1: Tecnologias empregadas nos servidores e comunicação entre eles.

## 4.2 Implementação no lado servidor

Após serem definidas quais as tecnologias são empregadas nos servidores e como eles interagem entre si, o próximo passo, para finalizar a configuração dos servidores, é definir o que precisou ser instalado e configurado para que a implementação dos sistemas web e *mobile* pudessem ser realizadas.

O *servidor OSN*, os *servidores de tiles* e o *serviço de pesquisa de locais e de geolocalização do Google Maps* não foram implementados neste trabalho, mas sim utilizados como parte da infraestrutura necessária para a implantação da solução, pois essas soluções já existiam, estavam consolidadas e, além disso, utilizá-las vai de encontro ao requisito não funcional de que se deve, quando possível, utilizar *frameworks* que auxiliem na padronização de projeto, reuso e coesão, aumentando a segurança e confiabilidade.

Diferentemente dos *servidores OSN* e de *tiles*, e do *serviço do Google Maps*, a infraestrutura OTP para a cidade de Porto Alegre não está disponível publicamente para utilização na web, portanto, para utilizá-la, se faz necessário instalar e configurar os

módulos OTP em um servidor específico, portanto, se fez necessário instalar e configurar um servidor para esse fim. A máquina utilizada para isso pertence ao Instituto de Informática da UFRGS e possui as seguintes configurações de hardware: Intel® Core™ i3 CPU 540 @ 3.07GHz x 2; memória RAM de 4GB; memória Principal (HD) de 500GB; e arquitetura de 64bits.

Levando em conta o requisito de que as soluções devem empregar, preferencialmente, software livre, então o sistema operacional escolhido e instalado nessa máquina foi o Ubuntu 13.10. Além da distribuição GNU/Linux, também foram instalados outros dois softwares para resolver as dependências do OTP: a *máquina virtual de desenvolvimento Java* (JDK), responsável por permitir a compilação do código fonte do OTP, o qual foi desenvolvido na linguagem de programação Java; e o *servidor web Tomcat*, responsável por disponibilizar o serviço web Java *OTP Routing API*.

Nessa mesma máquina em que os módulos OTP e o *Tomcat* foram instalados, também foi incluído e configurado o *servidor web Apache*. Esse servidor é o responsável por hospedar a aplicação web, a qual será detalhada na seção 4.3.1.

Com as ferramentas necessárias instaladas no servidor Ubuntu, a próxima etapa é a realização da configuração dos módulos OTP. Para gerar a base de dados que será consumida pelo *Routing*, o *GraphBuilder*, que é um utilitário de linha de comando, deve receber como entrada um arquivo de configuração no formato XML onde são fornecidos os dados cartográficos, bem como os dados e informações geográficas dos transportes públicos.

Os dados cartográficos utilizados no projeto foram obtidos através da ferramenta *Overpass API*, uma ferramenta web a qual permite o *download* de regiões específicas dos dados do mapa OSM no formato *OSM XML*. Essa ferramenta atua como um banco de dados na web onde o cliente envia uma requisição HTTP para a API informando os limites da região desejada e recebe como retorno o conjunto de dados no formato OSM XML contendo as informações solicitadas. Para obter os dados da região de Porto Alegre a seguinte requisição HTTP foi realizada: <http://www.overpass-api.de/api/xapi?map?bbox=-51.33911,-30.29227,-50.71564,-29.92637>. O arquivo recebido foi armazenado na mesma máquina onde o *GraphBuilder* está localizado para servir de fonte de dados para a geração do grafo.

Os dados e informações geográficas dos transportes públicos necessários para a geração do grafo devem estar especificados no formato *GTFS*, o qual possui um conjunto de arquivos contendo as paradas, as linhas, os horários, os itinerários e outras informações relevantes em arquivos texto com formatação específica que estão compactados em um único arquivo no formato ZIP. O conjunto completo de informações sobre os ônibus da cidade de Porto Alegre no formato GTFS foi obtido através do portal de dados abertos da cidade de Porto Alegre, o *DataPoa*<sup>15</sup>, por onde são disponibilizadas informações para permitir à comunidade desenvolver soluções que ajudem a cidade e seus moradores. Esse arquivo também foi armazenado na mesma máquina do *GraphBuilder* para completar as fontes de dados para geração do grafo.

Com o *OSM XML* e o *GTFS* adquiridos e armazenados no servidor, o *GraphBuilder* possui todas as informações necessárias para geração do grafo. Portanto, o arquivo de

---

<sup>15</sup> <http://www.datapoa.com.br/dataset/gtfs>

configuração XML deve ser atualizado apontando para as fontes de dados obtidas e o *GraphBuilder* deve ser executado para transformar os dados cartográficos e de transporte público em um grafo compilado armazenado em um único arquivo denominado *Graph.obj*, que será utilizado pelo *Routing* para a montagem dos caminhos.

A configuração do *Routing* também é feita através de um arquivo de configuração no formato XML. Nesse arquivo é especificado o caminho para o *Graph.obj* e também o algoritmo utilizado para calcular os trajetos. O cálculo dos caminhos foi configurado para utilizar o algoritmo A\* utilizando como heurística uma estratégia que leva em conta a distância Euclidiana e a velocidade de caminhada no cálculo de trechos a pé e a distância Euclidiana, velocidade, horário e custo de embarque para os trechos utilizando ônibus.

Para finalizar as configurações, o serviço web RESTful *OTP Routing API* foi hospedado no *servidor web Tomcat* para permitir a interação do OTP com as aplicações web e *mobile* recebendo requisições HTTP e retornando arquivos no formato JSON com os trajetos encontrados pelo *Routing*. Apesar de estar instalado no *Tomcat*, o *servidor web Apache* é quem será o responsável por receber as requisições ao *OTP Routing API* e repassá-las ao serviço hospedado no *Tomcat*. O *servidor web Tomcat* não será exposto externamente devido a restrições de segurança existentes.

Agora a estrutura e as ferramentas no lado servidor estão completas. Com base na Figura 4.1, a Figura 4.2 ilustra a configuração final do ViajeTriFácil, onde os servidores web e os módulo OTP foram instalados em uma única máquina. O próximo passo para completar a solução é desenvolver as aplicações web e *mobile* que interagirão com o usuário e comunicar-se-ão com os servidores.

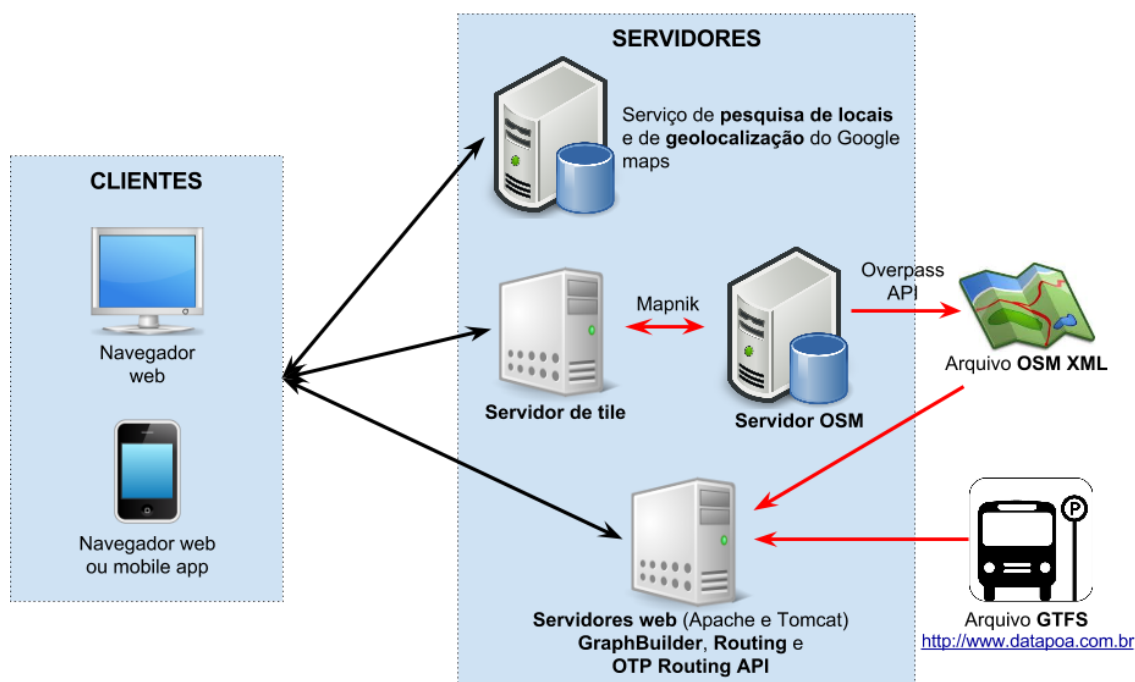


Figura 4.2: Tecnologias, comunicação, fonte de dados e estrutura física final dos servidores utilizados pelo ViajeTriFácil.

### 4.3 Implementação no lado cliente

Nesta seção são apresentadas as tecnologias empregadas para o desenvolvimento das soluções web e *mobile*, bem como algumas telas do sistema para ilustrar a aparência final da interface com o usuário. Como a solução final está fortemente baseada na especificação e no projeto, então não há necessidade de repetir todo o detalhamento do comportamento de cada uma das telas e ilustrá-las, pois tornaria a documentação extremamente repetitiva. Ao invés disso, em relação ao comportamento do sistema, é apresentado apenas algo específico que não esteja de acordo com o que foi definido.

#### 4.3.1 Sistema web

Para o desenvolvimento da interface web do ViajeTriFácil foram empregadas várias tecnologias para conseguir alcançar a solução ilustrada na Figura 4.3, onde é apresentada a tela principal da ferramenta vista através de um navegador web.

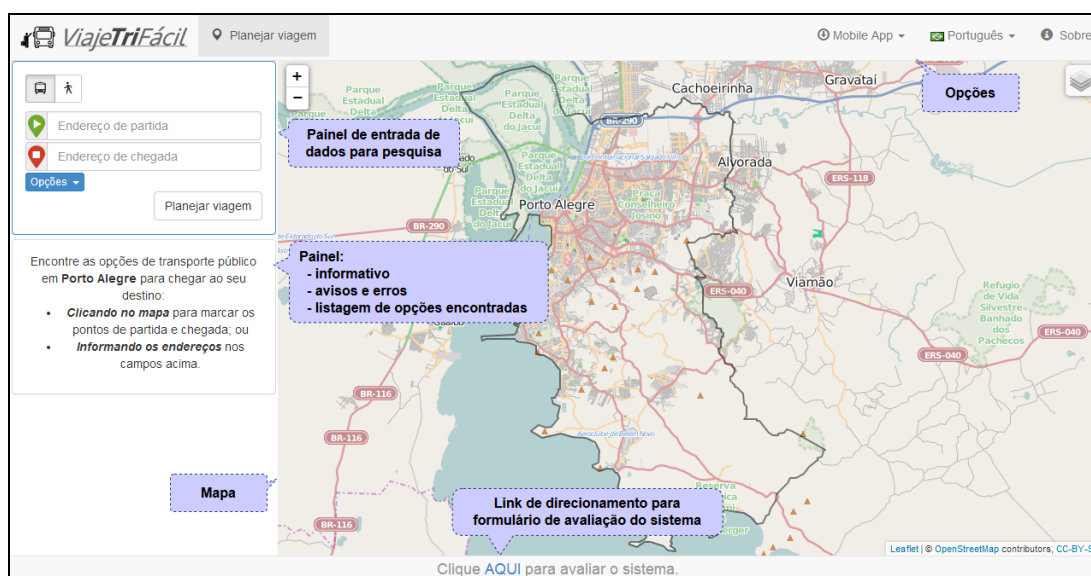


Figura 4.3: Interface web - tela de planejamento de viagens.

Para criar as páginas web é utilizado o **HTML5**, último padrão criado para o *HyperText Markup Language* (HTML), o qual foi projetado para gerar conteúdo rico na web sem a necessidade de instalação de *plug-ins* adicionais e para funcionar em PCs, tablets, smartphones ou smart TVs (HTML5 W3SCHOOLS, 2014).

Para controlar o estilo e o *layout* das páginas web é utilizado o **CSS3**, último padrão criado para o *Cascading Style Sheet* (CSS), o qual suporta todas as funcionalidades dos padrões anteriores e passou a ser dividido em módulos, permitindo que atualizações possam ser realizadas com maior frequência e que dispositivos com recursos reduzidos possam implementar apenas um subconjunto do CSS (CSS3 W3C, 2001).

O comportamento das páginas web é controlado por programas em **JavaScript**, uma linguagem de programação interpretada existente nos navegadores web que permite a execução de *scripts* no lado do cliente, sem a necessidade de passar pelo servidor (CROCKFORD, 2008).



Para facilitar, e simplificar, o uso de *scripts* no lado do cliente é utilizado o **jQuery**<sup>16</sup>, uma biblioteca JavaScript de código aberto desenvolvida para tornar mais simples a navegação sobre os documentos HTML, manipulação de eventos e criação de aplicações AJAX. Essa biblioteca, inclusive, é essencial para o uso de outras ferramentas utilizadas no projeto, as quais, sem ela, não funcionariam.

O *Asynchronous JavaScript and XML (AJAX)* é uma técnica no lado do cliente utilizada para trocar dados com um servidor e atualizar partes de uma página web de forma assíncrona, sem interferir no que o usuário está vendo enquanto a requisição está em andamento. Neste trabalho essa técnica é utilizada para preencher os valores no recurso de autocompletar das caixas de endereço, para atualizar o zoom e a área visível no mapa, e para buscar os caminhos ao planejar uma viagem (AJAX W3SCHOOLS, 2014).

A manipulação e formatação das datas em diferentes idiomas é feita utilizando o **Moment**<sup>17</sup>, uma biblioteca JavaScript de código aberto para converter, validar, manipular e formatar datas.

Para criar a aplicação web é utilizado o **Bootstrap**<sup>18</sup>, um *framework* de código aberto que utiliza HTML, CSS e JavaScript para criar um conjunto de componentes de interface permitindo a criação de páginas web simples, intuitivas e poderosas. Esse *framework* é utilizado no projeto, principalmente, por suportar uma interface web responsiva, o que vai de encontro com a especificação não funcional em que essa característica é requerida. Dessa forma, a interface web desenvolvida pode ser utilizada em um computador pessoal, um tablet ou mesmo em um smartphone, que a aparência da página se ajustará automaticamente, levando em conta as características do dispositivo utilizado. Sendo assim, com o uso dessa ferramenta é possível desenvolver a interface tanto da versão web quanto da versão *mobile* em um único código fonte.

Em conjunto com o Bootstrap são utilizados os *plug-ins* **Datepicker**<sup>19</sup> e **Timepicker**<sup>20</sup>, que atuam como componentes flutuantes na interface gráfica do usuário para facilitar a seleção da data e da hora, respectivamente.

A exibição e manipulação do mapa são realizadas através do **Leaflet**<sup>21</sup>, uma biblioteca de código aberto desenvolvida em Javascript para interagir com mapas com foco em simplicidade, desempenho e usabilidade que funciona tanto em computadores pessoais quanto em plataformas *mobile*. É essa biblioteca que faz a comunicação com o *tile server* para exibir as imagens (*tiles*) do mapa cartográfico em diferentes áreas e diferentes níveis de zoom. O ViajeTriFácil está configurado para poder utilizar um dos seguintes *tile servers*: *Open Street Map*; *OSM CloudMade*; *Open Cycle Map*; ou *OSM MapQuest*. Desses *tile servers* citados, o *Open Street Map* é o padrão, mas pode ser alterado a qualquer momento no ícone existente no canto superior direito do mapa.

---

<sup>16</sup> <http://jquery.com>

<sup>17</sup> <http://momentjs.com>

<sup>18</sup> <http://getbootstrap.com>

<sup>19</sup> <http://eternicode.github.io/bootstrap-datepicker>

<sup>20</sup> <http://jdewit.github.io/bootstrap-timepicker>

<sup>21</sup> <http://leafletjs.com>

Em conjunto com Leaflet são utilizados alguns *plug-ins*. O **Context Menu**<sup>22</sup>, que possibilita a geração de um menu de contexto sobre o mapa, o qual permite ao usuário executar as funções especificadas na prototipação da interface web quando o usuário clicar sobre o mapa com o botão direito do mouse. O **Awesome Markers**<sup>23</sup>, que possibilita a criação de ícones personalizados sobre o mapa. E, por fim, o **Encoded**<sup>24</sup>, que é responsável por decodificar as coordenadas geográficas de cada um dos trajetos recebidas do *OTP Routing API* para possibilitar a exibição delas no mapa.

Para o recurso de autocompletar das caixas de texto utilizadas para informar os endereços de origem e destino é utilizado o **Google Maps Javascript API v3**<sup>25</sup>, uma biblioteca Javascript gratuita, mas de código proprietário, disponibilizada pela empresa Google que fornece uma série de utilitários para manipulação de mapas. Essa biblioteca faz chamadas a um serviço de **pesquisa de locais** responsável por receber um texto e retornar previsões de locais com descrição similar ao texto recebido.

Para o preenchimento, no mapa, das coordenadas (latitude e longitude) de um determinado endereço informado nas caixas de texto, ou vice-versa, também é utilizada a mesma biblioteca do Google, porém o serviço utilizado é o de **geocodificação**, um serviço assíncrono que faz chamadas para o servidor de geocodificação para converter endereços em coordenadas geográficas e vice-versa.

A exibição dos dados recuperados do *OTP Routing API* e a troca automática do idioma do sistema sem qualquer recarregamento da página é feita utilizando o **Knockout**<sup>26</sup>, uma biblioteca Javascript que permite separar claramente o modelo de dados da apresentação das informações, além de atualizar automaticamente a interface quanto o estado de alguma informação no modelo de dados é alterado.

Portanto, com todas as tecnologias empregadas no sistema web já definidas, agora fica mais fácil entender como é feito o planejamento de uma viagem. Quando o usuário clica no botão responsável por planejar a viagem é feita uma chamada AJAX através da biblioteca *jQuery* ao *OTP Routing API*, o qual retorna um arquivo no formato JSON e este é utilizado para atualizar o modelo de dados dos trajetos exibidos na interface do usuário. Quando esse modelo é alterado o *Knockout* entra em ação e atualiza automaticamente a interface do usuário com essas novas informações, pois o estado dos dados foi modificado. A Figura 4.4 ilustra o resultado de uma pesquisa entre dois pontos distintos e a exibição dos detalhes do trajeto.

---

<sup>22</sup> <https://github.com/aratcliffe/Leaflet.contextmenu>

<sup>23</sup> <https://github.com/lvoogdt/Leaflet.awesome-markers>

<sup>24</sup> <https://github.com/jieter/Leaflet.encoded>

<sup>25</sup> <https://developers.google.com/maps/documentation/javascript>

<sup>26</sup> <http://knockoutjs.com>

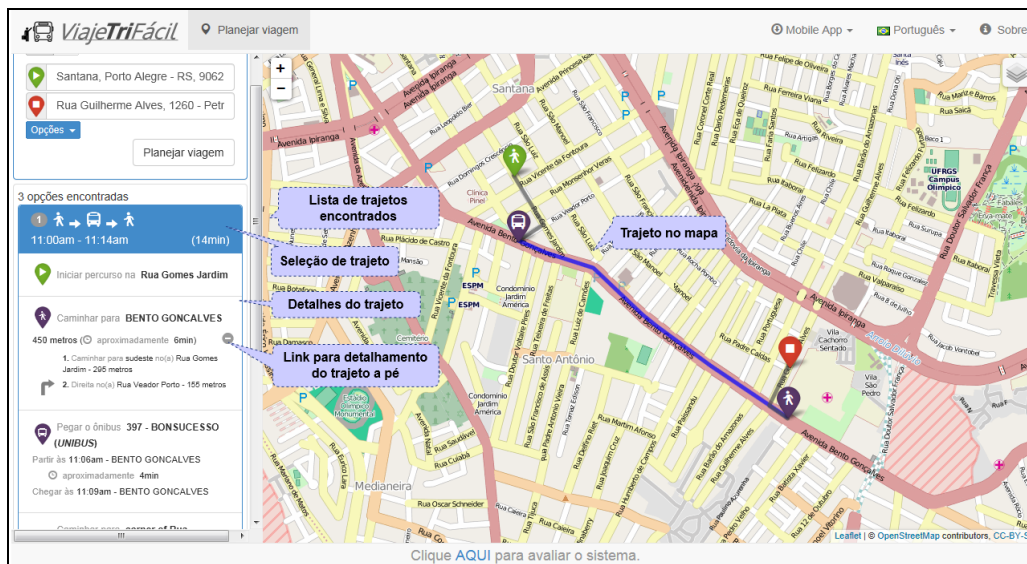


Figura 4.4: Interface web – listagem e exibição de trajetos.

Da mesma forma que as informações de trajetos são atualizadas, ao alterar o idioma da aplicação o *Knockout* identifica que a fonte de dados com os textos exibidos na tela foi alterada e, automaticamente, atualiza as informações na interface sem realizar o recarregamento da página web.

Com o uso das tecnologias de desenvolvimento apresentadas foi possível construir a interface web proposta neste trabalho atendendo a grande maioria das funcionalidades especificadas, com exceção de algumas. As exceções são: (i) as cores de fundo de cada um dos trajetos exibidos na listagem não são diferenciadas, permanecendo a cor de fundo cinza para todos os trajetos da lista, havendo diferenciação apenas para o trajeto selecionado, o qual aparece com fundo azul; (ii) a exibição dos trechos percorridos através de ônibus em cada um dos trajetos exibidos na listagem não apresenta o código da linha, mas sim o símbolo de um ônibus; e, por fim, (iii) a exibição do mapa e dos detalhes do trajeto não são exibidas apenas quando é requisitado pelo usuário, mas sim sempre que uma pesquisa é realizada, onde são apresentadas as informações do primeiro trajeto. A Figura 4.5 ilustra a tela contendo essas diferenças.

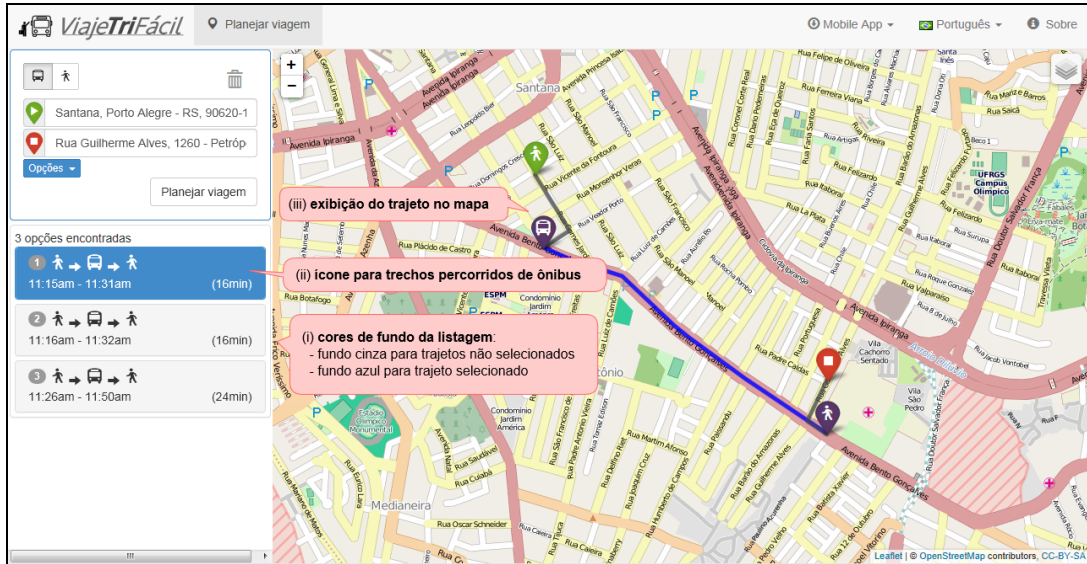


Figura 4.5: Interface web – diferenças em relação ao que foi especificado.

### 4.3.2 Sistema mobile

O desenvolvimento da interface para dispositivos móveis foi realizado com as mesmas tecnologias empregadas para o desenvolvimento da interface web, pois, com o uso da biblioteca **Bootstrap** foi possível desenvolver um único sistema capaz de se adaptar ao dispositivo no qual está em execução, tendo aparências diferentes quando executado em um computador pessoal, um tablet ou um smartphone. Nas Figuras 4.6 e 4.7 são ilustradas, respectivamente, as interfaces da tela de planejamento de viagens e a lista de trajetos, as quais são acessadas pelo navegador web nativo de um smartphone.



Figura 4.6: Interface mobile – tela de planejamento de viagens.

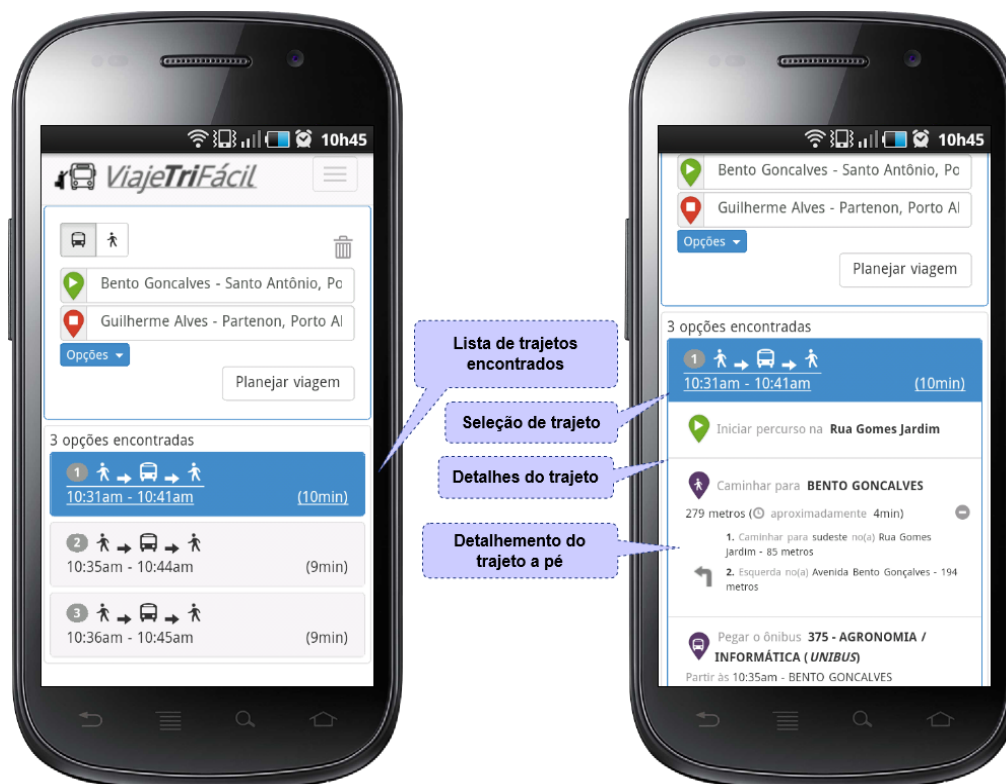


Figura 4.7: Interface *mobile* – lista e exibição de trajetos.

Para criar o aplicativo *mobile*, o qual pode ser instalado no dispositivo do usuário, não foram utilizadas linguagens nativas de alguma plataforma específica, mas sim o **PhoneGap**, um *framework* gratuito e de código aberto que permite a criação de aplicativos para diversas plataformas de dispositivos móveis a partir de aplicações construídas utilizando *HTML*, *CSS* e *JavaScript*. Essa ferramenta é capaz de encapsular o aplicativo baseado em tecnologias web e transformá-lo em um aplicativo para ser distribuído, instalado e ter acesso aos recursos nativos das APIs do dispositivo. O resultado final é um aplicativo híbrido que não é totalmente nativo e nem totalmente web (ATRICE, 2012).

Na Figura 4.8 está ilustrado o processo de geração dos pacotes para diferentes plataformas a partir do *PhoneGap*. Através dele é possível gerar aplicações que podem ser distribuídas para diversas plataformas *mobile*, tais como iOS, Android, webOS, Windows Phone e Symbian OS e BlackBerry. Para o ViajeTriFácil foram geradas aplicações para as plataformas Android, webOS, Windows Phone e Symbian, as quais possuem a mesma aparência das aplicações web e podem ser instaladas a partir do *download* do pacote diretamente da interface web no link “*Mobile App*” apresentado na Figura 4.6. A aplicação não foi disponibilizada para iOS, pois, para gerá-la, seria necessário ter um Apple ID e, além disso, pagar a ativar uma conta de desenvolvedor.

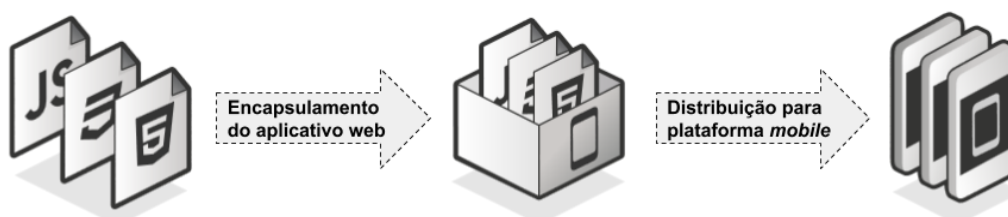


Figura 4.8: Geração de aplicações *mobile* com *PhoneGap* (<http://phonegap.com/about>).

Da mesma forma que a aplicação web, as mesmas exceções de comportamento e aparência em relação ao que foi especificado também são vistas na interface para dispositivos móveis. Além dessas, há uma diferença em relação ao que foi especificado para a interface de dispositivos móveis, que é a interação com o mapa. Na solução construída não há interação com mapa, seja para marcar pontos, ou para exibição de trajetos, conforme pode ser visto na Figura 4.7. Essa diferença se deve ao fato do foco principal de desenvolvimento ter sido direcionado à interface para computadores pessoais, e não para dispositivos móveis. A implementação para dispositivos móveis envolve uma série de detalhes que são dependentes do tipo de dispositivo móvel e do seu sistema operacional nativo. Isso despenderia um importante esforço adicional de programação que desviaria dos objetivos deste trabalho.

#### 4.4 Considerações finais

Neste capítulo foi apresentada a implementação do sistema ViajeTriFácil através da definição das ferramentas e da estrutura física utilizada no lado servidor, com foco no uso de tecnologias de código aberto; da instalação e configuração dos módulos OTP; das fontes de dados utilizadas para alimentar o *Routing*, módulo OTP responsável pela realização do cálculo de trajetos; das tecnologias empregadas no desenvolvimento das soluções web e *mobile*; da forma de comunicação entre clientes e servidores; e também das diferenças existentes em relação ao que foi especificado.

Uma dessas diferenças está relacionada aos requisitos não funcionais, pois, para ser possível completar o desenvolvimento da aplicação com tudo que foi especificado, foi necessário utilizar os serviços de preenchimento automático e de geocodificação do Google, os quais não são software livre. Esses recursos possuem seu uso limitado à realização de 25.000 requisições por dia e, caso este limite seja excedido por mais de 90 dias consecutivos, então será necessário pagar pelas requisições excedentes.

Outra característica que pode ser considerada divergente em relação aos requisitos não funcionais é o fato de não ter sido desenvolvida uma solução nativa para dispositivos móveis, mas sim utilizado uma ferramenta, o PhoneGap, responsável por transformar uma aplicação web em uma aplicação *mobile*. Apesar disso, através dessa ferramenta foi possível gerar versões para diversos dispositivos móveis, o que seria inviável se fosse necessário desenvolver uma aplicação nativa para cada uma das plataformas.

No próximo capítulo é apresentada a última etapa da construção do software proposto neste trabalho, a validação. Nessa etapa serão apresentados alguns testes básicos das funcionalidades disponíveis no sistema e também uma boa visão, do ponto de vista do usuário, sobre alguns aspectos do ViajeTriFácil como a corretude, a usabilidade e o desempenho da ferramenta.

## 5 VALIDAÇÃO

Com a solução implementada, o próximo passo é realizar a avaliação do sistema para verificar se ele atende a aspectos como corretude, usabilidade e desempenho. Portanto, este capítulo traz a aplicação e o resultado de avaliações realizadas sobre o ViajeTriFácil feitas através de testes sobre a ferramenta e da aplicação de uma pesquisa de opinião aos usuários finais.

### 5.1 Metodologia

A avaliação da ferramenta foi feita através de dois métodos distintos. O primeiro foi a realização de testes exploratórios sobre as funcionalidades básicas disponibilizadas pela ferramenta. O segundo método consistiu em uma pesquisa de opinião feita através de um questionário para avaliar a visão do usuário em relação ao que o ViajeTriFácil oferece.

A realização de testes funcionais básicos foi feita com a aplicação de alguns cenários de testes, tais como, a viagem de um ponto “A” a um ponto ”B” pegando apenas um ônibus e mais de um ônibus. Para avaliar a corretude dos resultados apresentados foi empregado o conhecimento prévio do trajeto informado, onde o resultado esperado já é conhecido, e comparado com o resultado apresentado pela ferramenta.

A realização da pesquisa de opinião consistiu na aplicação de um questionário com algumas perguntas desenvolvidas para avaliar o perfil do usuário, a usabilidade, a corretude e o desempenho da ferramenta. Esse questionário é apresentado no Apêndice deste trabalho. O questionário foi criado através do *Google Forms*, uma ferramenta disponibilizada pela empresa Google que permite a criação de um formulário e a coleta das respostas dos usuários na Internet, as quais são armazenadas em uma planilha e também utilizadas para a apresentação gráfica resumida de todas as avaliações. A forma utilizada para disponibilizar o questionário para os usuários foi através da inserção de um *link* na página principal do ViajeTriFácil, o qual pode ser visualizado no rodapé da página em computadores pessoais (Figura 4.3), ou no menu de opções em dispositivos móveis (Figura 4.6).

### 5.2 Testes funcionais básicos

Os testes funcionais consistiram na avaliação de alguns cenários de testes realizados sobre a funcionalidade principal da ferramenta, que é o planejamento de viagem, feitos através da comparação entre o resultado esperado com o resultado apresentado. Os

cenários escolhidos levam em conta a quantidade de ônibus necessária para se deslocar de um ponto “A” até um ponto “B”. Em alguns casos, foi considerada a variação de parâmetros opcionais como, por exemplo, data de partida e distância a percorrer a pé, para apresentar a sensibilidade da busca quanto à variação deles.

### 5.2.1 Cenário I: Um ônibus do ponto “A” ao ponto “B”

O primeiro cenário de teste avaliado foi o trajeto do Campus do Centro da UFRGS, na Avenida Paulo Gama, 110 até a PUCRS, na Avenida Ipiranga, 6681. O trajeto foi consultado em uma sexta-feira às 16h15min, sem variar qualquer outro parâmetro opcional. O sistema apresentou como resultado as possibilidades de utilizar os ônibus D43 ou 353, na Avenida Osvaldo Aranha, o que vai de acordo com os possíveis resultados esperados. O resultado é ilustrado na Figura 5.1.

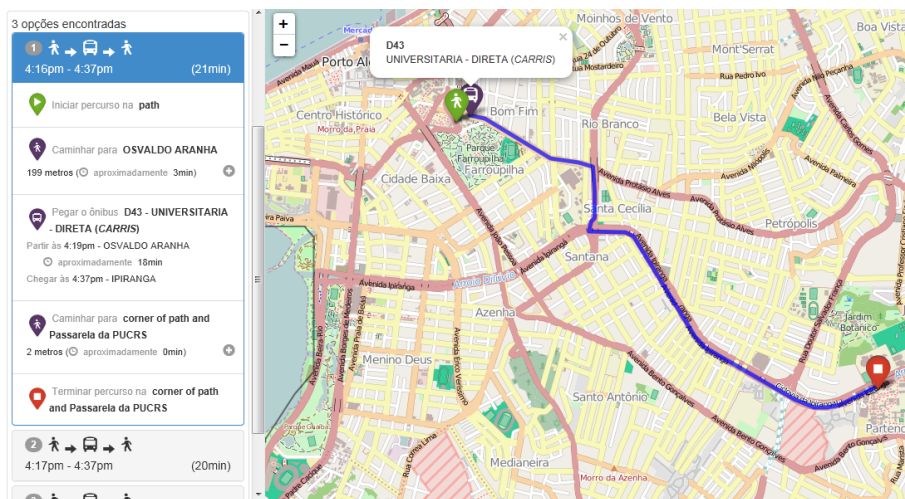


Figura 5.1: Teste funcional – Campus Centro UFRGS até PUCRS.

Para avaliar a sensibilidade quanto à data, o próximo teste utilizou os mesmos pontos de origem e destino, porém a data de partida foi alterada para um sábado no mesmo horário, às 16h15min. Conforme esperado, o resultado apresentado foi alterado, exibindo como possibilidade os ônibus 353 e 4763, não exibindo o ônibus D43, pois seu funcionamento é apenas nos dias úteis e no período matutino aos sábados. O resultado é ilustrado na Figura 5.2.

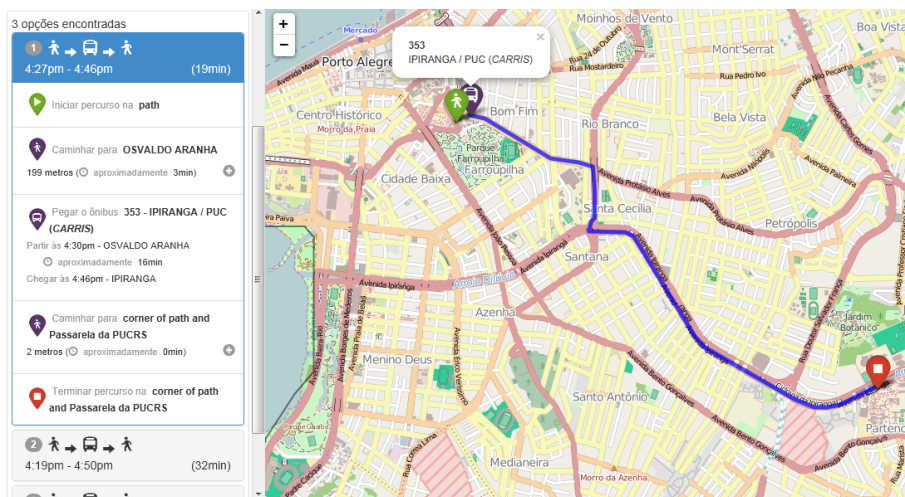


Figura 5.2: Teste funcional – Campus Centro UFRGS até PUCRS alterando data.



Para avaliar a sensibilidade de hora, o mesmo trajeto dos dois testes anteriores foi utilizado, deixando a data em um sábado e alterando a hora das 16h15min para às 10h15min. O resultado apresentou como possibilidades os ônibus 353, D43 e 4763, mostrando que a sensibilidade quanto à hora está funcionando corretamente, pois o ônibus D43 foi exibido, conforme ilustra a Figura 5.3.

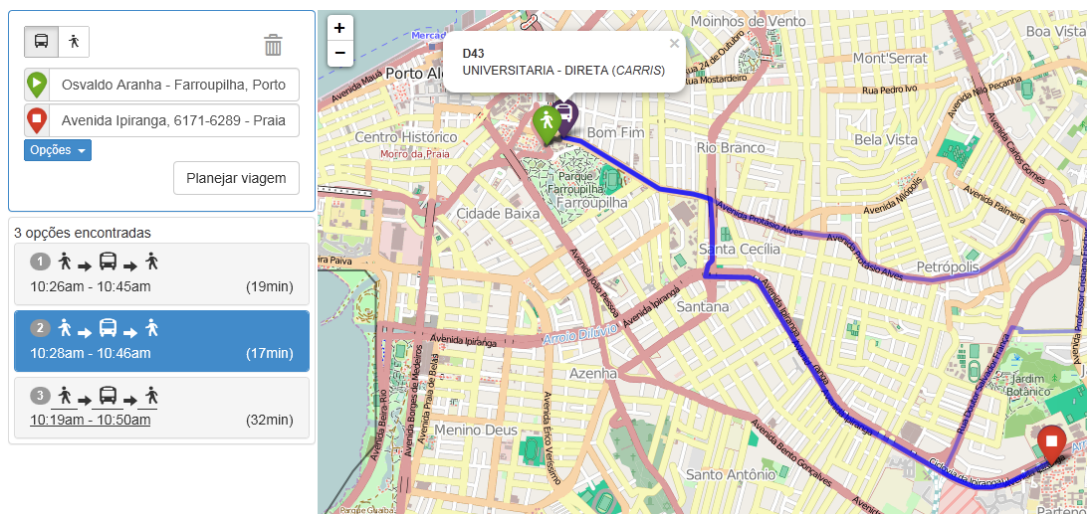


Figura 5.3: Teste funcional – Campus Centro UFRGS até PUCRS alterando hora.

Para mostrar um trajeto em que a parada de desembarque apresentada está incorreta foi avaliado o trajeto do Campus do Centro da UFRGS até o Instituto de Informática, no Campus do Vale da UFRGS. Conforme esperado, a ferramenta apresentou as opções de ônibus corretas (343 ou D43), contudo, não apresentou o melhor local para descer do ônibus, que seria exatamente no endereço de destino ao invés do local indicado na Figura 5.4. O motivo pelo qual o sistema informa para o usuário descer na parada incorreta é que a especificação GTFS para os ônibus 343 e D43 consideram que não existe uma parada no Instituto de Informática da UFRGS.

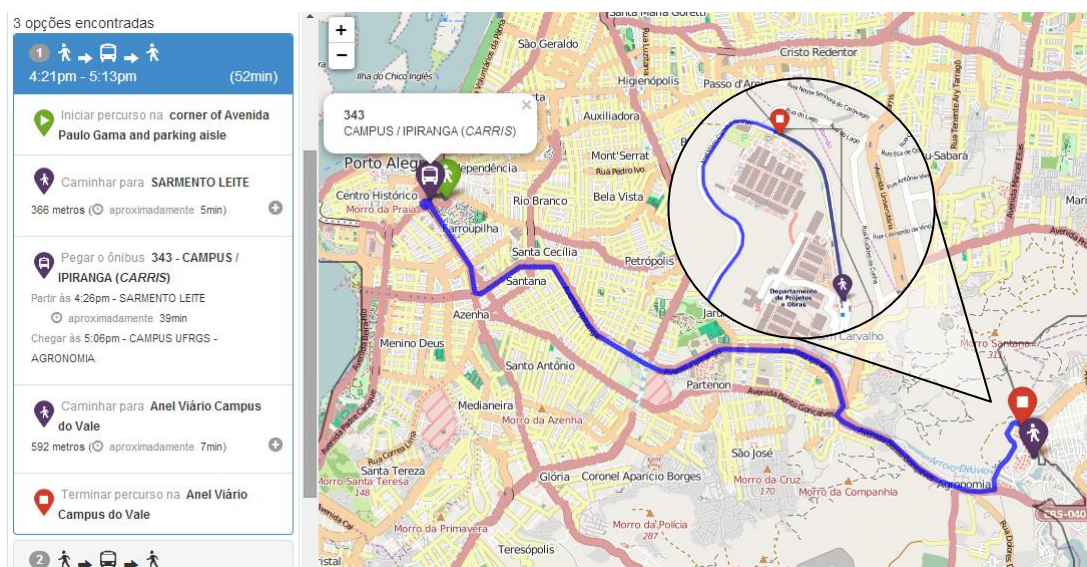


Figura 5.4: Teste funcional – Parada de desembarque incorreta no trajeto do Campus Centro UFRGS até Campus Vale UFRGS.

Para concluir os testes funcionais considerando a utilização de um único ônibus foi avaliado um cenário de teste em que as linhas de ônibus apresentadas como opção estão corretas, porém o trajeto desenhado no mapa está incorreto. Os pontos de origem e destino escolhidos foram, respectivamente, na Avenida Severo Dullius, 2150, e na Avenida Farrapos, 2900. O resultado da consulta apresenta as opções de ônibus corretas (7051 ou B56), porém, o desenho do trajeto do ônibus 7051 no mapa está inconsistente, apresentando os caminhos de ida e de volta do ônibus 7051 entre os endereços de origem e destino, conforme ilustra a Figura 5.5. O motivo provável porque isso acontece é um problema existente no algoritmo que realiza o desenho do trajeto quando aplicado sobre linhas de ônibus que possuem trechos que se sobrepõem.

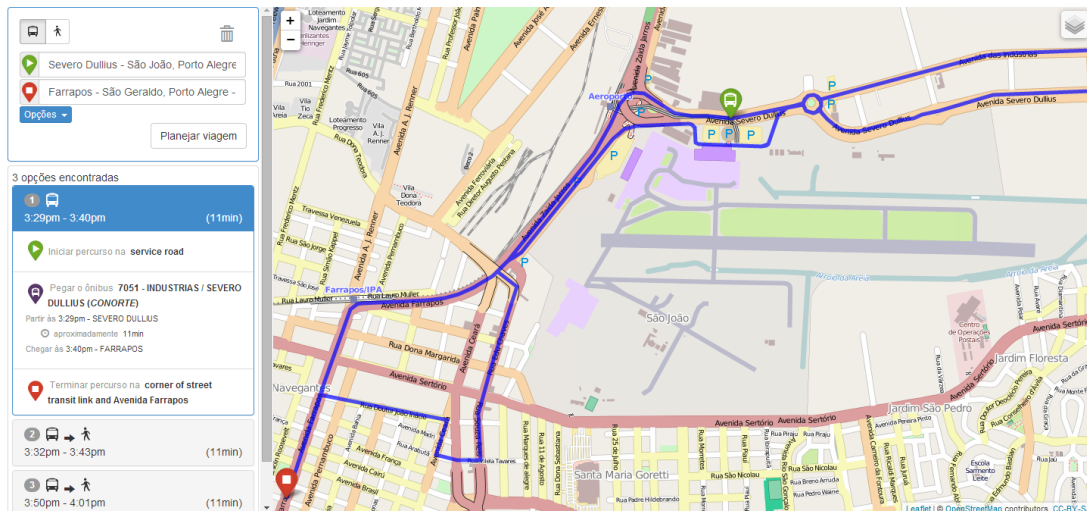


Figura 5.5: Teste funcional – Rua Miguel Tostes até Av. Borges de Medeiros considerando 1km de percurso máximo a pé

## 5.2.2 Cenário II: Dois ônibus do ponto “A” ao ponto “B”

O primeiro cenário de teste utilizado para avaliar um percurso considerando dois ônibus utiliza como endereço de partida o Shopping Praia de Belas, na Avenida Ipiranga e como endereço de chegada o Campus do Vale da UFRGS. O resultado da pesquisa apresentou três combinações distintas de ônibus: 264 e 375; T1 e 343; e T1 e D43. Pelo conhecimento prévio do caminho, os resultados apresentados pela ferramenta estão corretos. A Figura 5.6 apresenta o resultado da pesquisa com a combinação dos ônibus T1 e D43 selecionada.

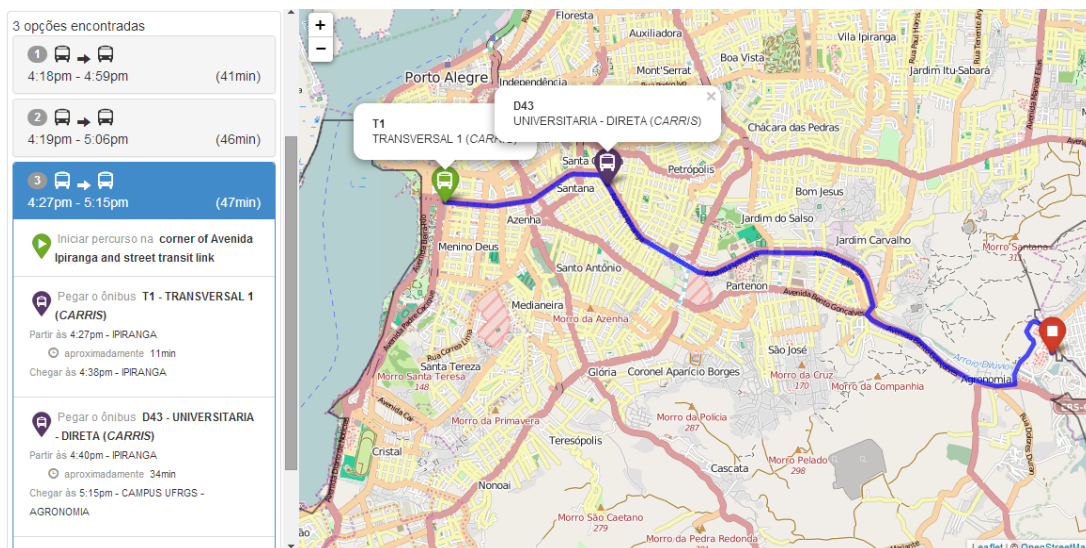


Figura 5.6: Teste funcional – Shopping Praia de Belas até PUCRS.

O próximo teste considerando dois ônibus tem a intenção de mostrar uma integração em que o passageiro precisa, além de pegar dois ônibus, percorrer um trecho a pé para pegar o segundo ônibus. O percurso escolhido tem como partida o Campus da Saúde da UFRGS, na Avenida Ipiranga e como destino o Estádio Beira Rio, na Avenida Padre Cacique. Em todas as opções apresentadas pelo sistema (T1 e 149; T1 e 173; T1 e 184) é necessário pegar dois ônibus e, para pegar o segundo, caminhar a pé, o que está de acordo com as opções conhecidas. A Figura 5.7 apresenta a combinação dos ônibus T1 e 149, com zoom no trecho em que é necessário realizar a integração.

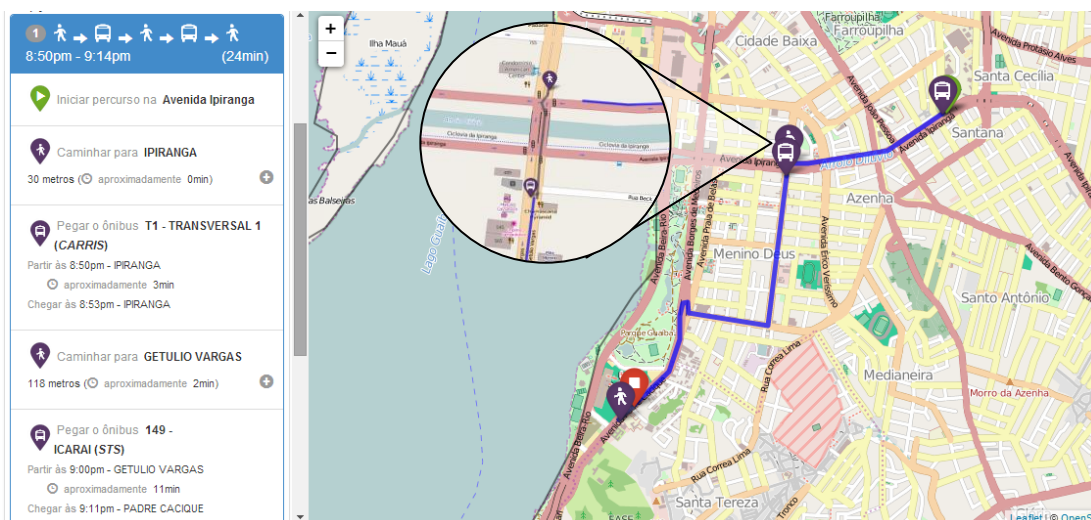


Figura 5.7: Teste funcional – Campus Saúde UFRGS até Estádio Beira Rio.

O último teste considerando dois ônibus mostra a sensibilidade quanto à variação da distância percorrida a pé. Para avaliar essa sensibilidade foi considerado um percurso com partida da Rua Miguel Tostes, 900, até o prédio do IPE, na Avenida Borges de Medeiros, 1945. Ao considerar a distância máxima padrão para caminhada de 500m o sistema sempre apresenta como resultado trajetos com combinação de dois ônibus (T7 e C1; T7 e 179; T7 e 184), conforme ilustrado na Figura 5.8. Porém, pelo conhecimento prévio do caminho, essa não seria a melhor opção, visto que o trajeto pode ser percorrido utilizando apenas um ônibus (apenas o T7). O motivo pelo qual o sistema

indica dois, ao invés de um ônibus, é que a distância necessária a ser percorrida a pé ao utilizar apenas um ônibus é de aproximadamente 550m, ou seja, um valor superior à opção padrão configurada, que é de, no máximo, 500m. Para que o sistema indique apenas um ônibus a distância máxima a pé precisou ser alterada para 1km, fazendo com que o sistema faça as sugestões esperadas, conforme apresentado na Figura 5.9.

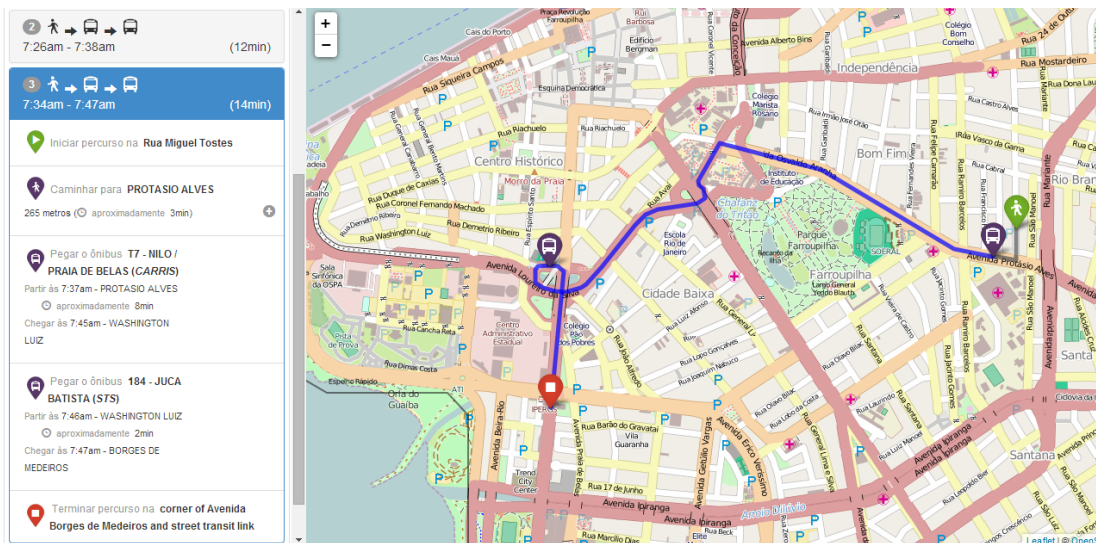


Figura 5.8: Teste funcional – Rua Miguel Tostes até Av. Borges de Medeiros considerando 500m de percurso máximo a pé.

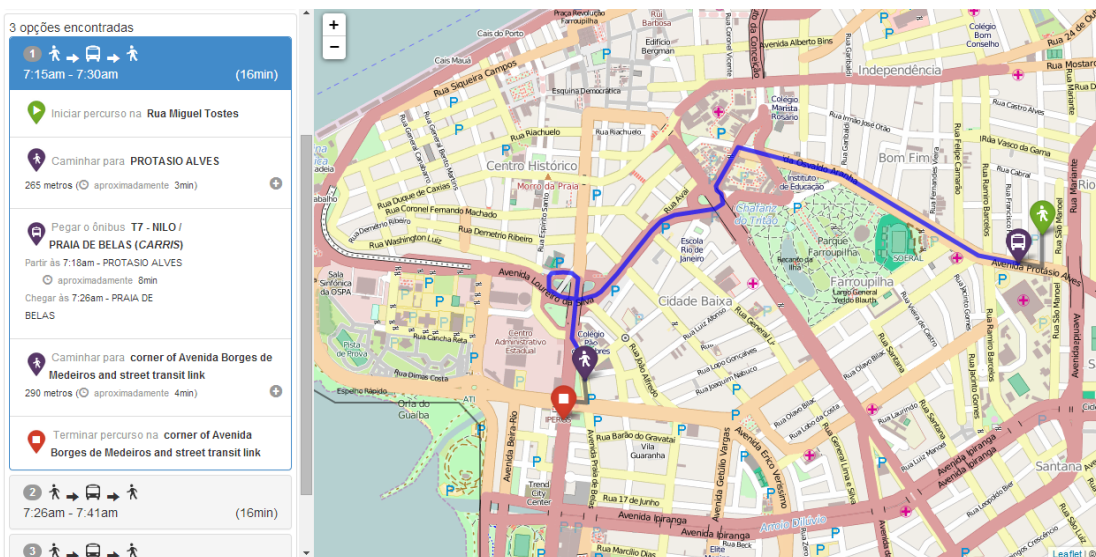


Figura 5.9: Teste funcional – Rua Miguel Tostes até Av. Borges de Medeiros considerando 1km de percurso máximo a pé.

### 5.2.3 Cenário III: Três ônibus do ponto “A” ao ponto “B”

Para demonstrar um percurso em que são utilizados três ônibus foi escolhido como ponto de partida a Avenida Bento Gonçalves, 4500, próximo à PUCRS, e como ponto de chegada a FIERGS, na Avenida Assis Brasil, 8787. O resultado da ferramenta apresentou três opções com combinações distintas (R321, 731 e D72; R32, 756 e D73; 397, 431 e D72), onde a primeira delas é apresentada na Figura 5.10.

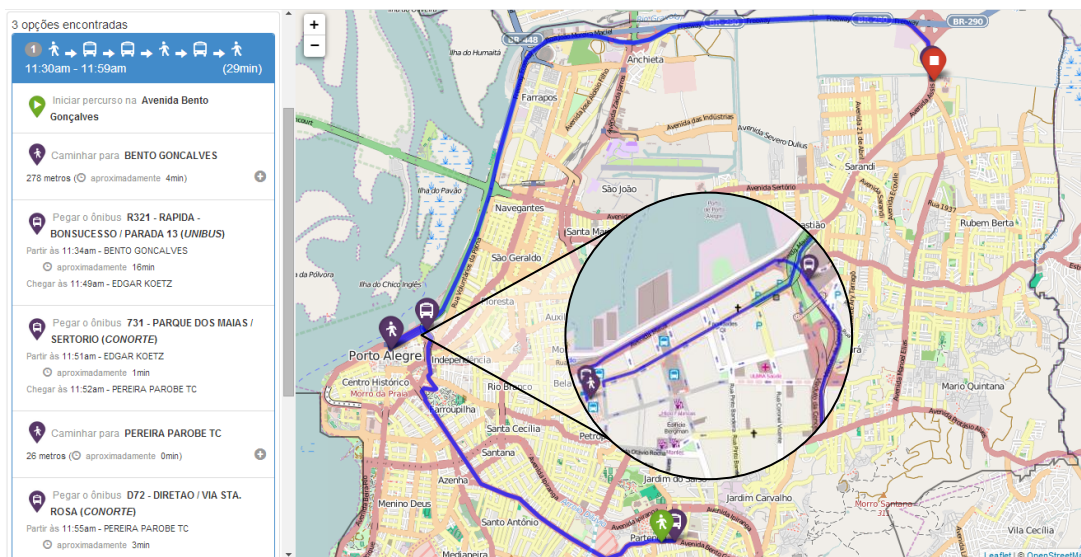


Figura 5.10: Teste funcional – Av. Bento Gonçalves, 4500, até FIERGS considerando 500m de percurso máximo a pé.

Da mesma forma que o último teste do cenário II, onde o trajeto poderia ser otimizado para apenas um ônibus alterando a distância máxima percorrida a pé, esse trajeto também pode ser otimizado para utilizar dois ônibus, ao invés de três, alterando o percurso máximo a pé para 1km, conforme ilustra a Figura 5.11.

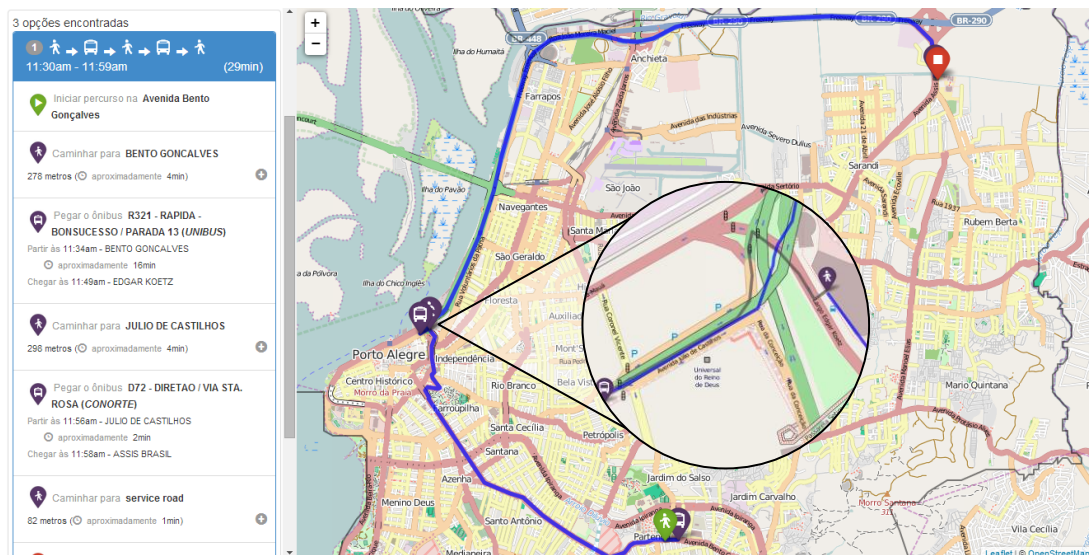


Figura 5.11: Teste funcional – Av. Bento Gonçalves, 4500, até FIERGS considerando 1km de percurso máximo a pé.

### 5.3 Considerações parciais

Ao realizar os testes exploratórios sobre o ViajeTriFácil foi possível chegar a algumas conclusões em relação às funcionalidades que ela oferece. A ferramenta seleciona opções corretas de linhas de ônibus, responde a sensibilidade na alteração dos parâmetros opcionais de data, hora e distância máxima a percorrer a pé, e retorna resultados considerados extremamente satisfatórios, apesar de erro no desenho de algumas linhas.

Testes sobre trajetos com quatro ou mais ônibus não foram realizados, pois a ferramenta foi limitada ao máximo de três para evitar queda de desempenho no algoritmo de busca. Além disso, trajetos com mais de três ônibus é uma situação incomum a qual não se tem conhecimento de caminhos em Porto Alegre em que quatro ônibus seriam necessários. Até mesmo um trajeto com três ônibus é uma situação difícil de encontrar na cidade.

## 5.4 Pesquisa de opinião

O questionário utilizado para avaliar o perfil do usuário, a usabilidade, a corretude e o desempenho da ferramenta foi disponibilizado através de um *link* no ViajeTriFácil e utilizou a ferramenta *Google Forms*. Os dados aqui exibidos representam a opinião de 129 usuários do sistema, obtidos no período compreendido entre sete de maio de 2014 e dezessete de junho de 2014.

### 5.4.1 Perfil do usuário

O perfil do usuário que respondeu ao questionário de avaliação foi feito através de três perguntas objetivas simples para indicar o sexo, a idade e a escolaridade, respectivamente.

Ao analisarmos o resultado da pesquisa na Figura 5.12 é possível observar que a grande maioria dos usuários que respondeu o questionário é do sexo masculino (77%), devido ao fato da maior divulgação da ferramenta ter sido direcionada ao Instituto de Informática da UFRGS, onde a maioria dos estudantes é do sexo masculino.

Também é notável que quem avaliou sobre a ferramenta têm entre 19 e 45 anos (78%) e possui ensino superior ou pós-graduação (85%), seja completo ou em andamento, indo de encontro ao público alvo da pesquisa, que foram estudantes universitários da UFRGS, os quais possuem a idade média e o nível de escolaridade da maioria das avaliações.

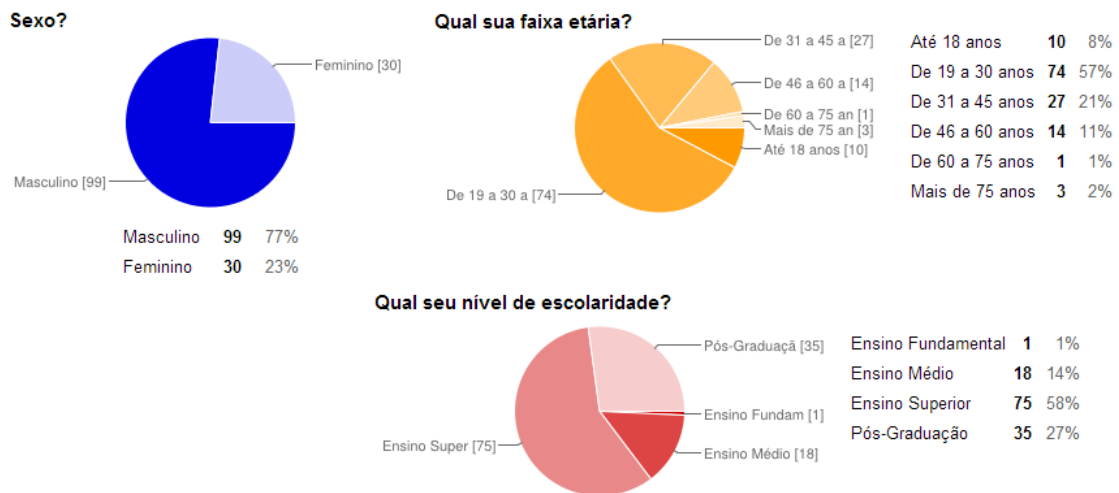


Figura 5.12: Pesquisa de opinião – perfil do usuário.

## 5.4.2 Mobilidade

As questões de mobilidade avaliaram a frequência de uso de ônibus e quais os principais problemas no transporte público de Porto Alegre. O resultado é apresentado na Figura 5.13.

A pesquisa mostrou que, das 129 pessoas que responderam ao formulário, 88 delas (68%) utilizam ônibus quatro ou mais vezes por semana, indicando que quem preencheu o questionário é usuário frequente dessa modalidade de transporte e, portanto, a resposta dessas pessoas tem significância por terem conhecimento de como funciona o transporte público da cidade.

Na avaliação dos problemas existentes foram apontados quatro deles como principais, onde 75% dos entrevistados apontaram a falta de informação como um dos problemas, 81% indicaram a superlotação, 58% indicaram o congestionamento e 71% consideraram a pontualidade. Essa avaliação mostra que a falta de informação é um dos grandes problemas no transporte público da cidade, justificando a necessidade de disponibilizar informações sobre trajetos e itinerários para os usuários. Os outros três problemas em destaque não podem ser resolvidos apenas com ações da comunidade, é necessária a cooperação entre permissionárias de ônibus e entidades governamentais para realização de ações, tais como, melhorias viárias, priorização do transporte público em detrimento do individual e aumento na frota de ônibus.

Além dos problemas sugeridos no questionário, 11 pessoas indicaram outros problemas distintos com o transporte público, tais como, segurança, informação em tempo real da localização dos ônibus, poluição sonora, preço, treinamento de motoristas e cobradores, informação sobre localização de paradas, poucos horários e falta de recursos para disponibilização de itinerários.



Figura 5.13: Pesquisa de opinião – mobilidade.

### 5.4.3 Software de mobilidade urbana

As questões sobre softwares de mobilidade urbana avaliaram a opinião do usuário sobre o uso da tecnologia como forma de disponibilizar informações sobre trajetos e itinerários.

Conforme ilustrado na Figura 5.14, a pesquisa mostrou que a grande maioria dos usuários (98%) indicou a Internet e as redes sociais como o meio ideal para disponibilizar informações sobre trajetos e itinerários, e que quase todos já utilizaram algum sistema para encontrar meios de transporte como o site da EPTC, o *Google Transit*, o *PoaBus* e o *PoaTransporte*. Além das ferramentas sugeridas, algumas pessoas indicaram outras, tais como *MovelBus*<sup>27</sup>, *DB Navigator*<sup>28</sup>, *Transport Sydney Buses*<sup>29</sup> e *Trafegue Bem*<sup>30</sup>.

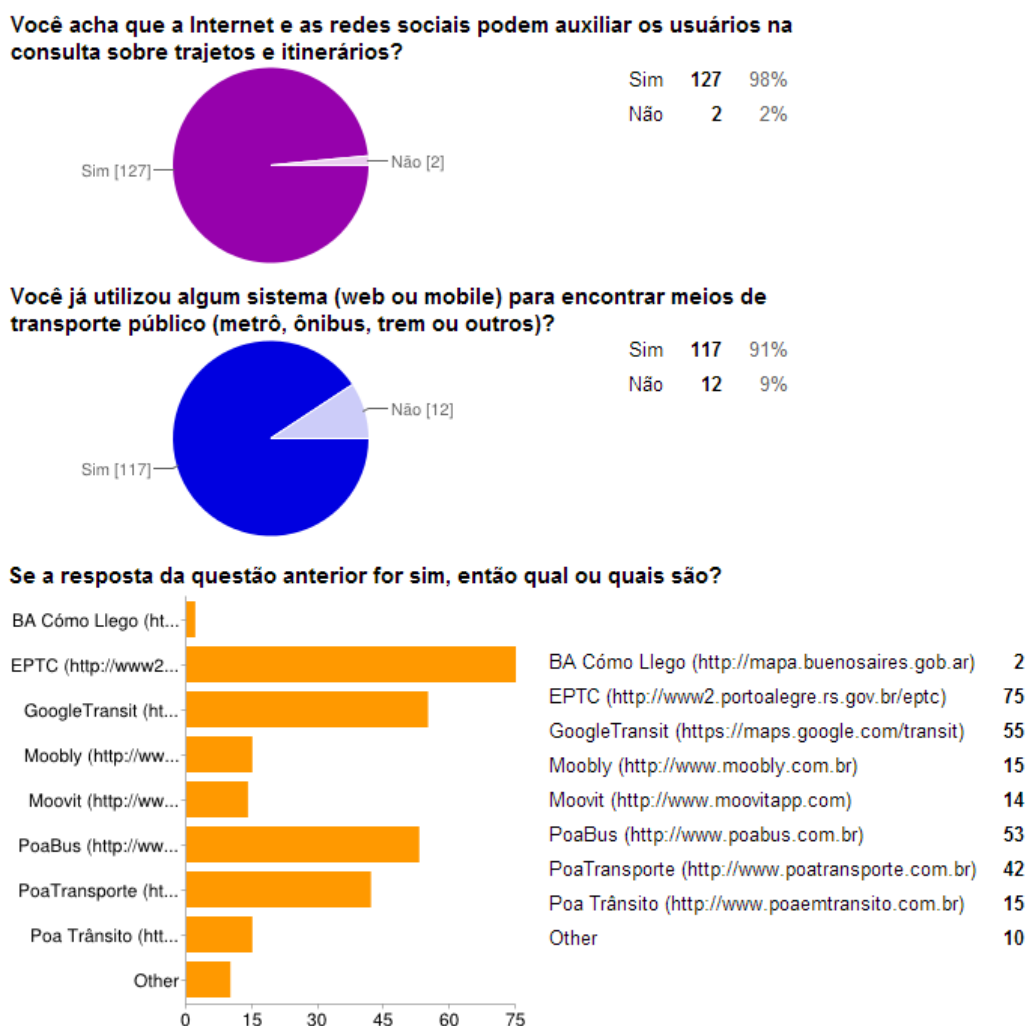


Figura 5.14: Pesquisa de opinião – softwares de mobilidade urbana.

<sup>27</sup> <http://www.movelbus.com.br/>

<sup>28</sup> <http://www.bahn.de/p/view/buchung/mobil/db-navigator.shtml>

<sup>29</sup> <http://www.sydneybuses.info/routes/timetables-route-maps>

<sup>30</sup> <http://www.trafeguebem.com.br>

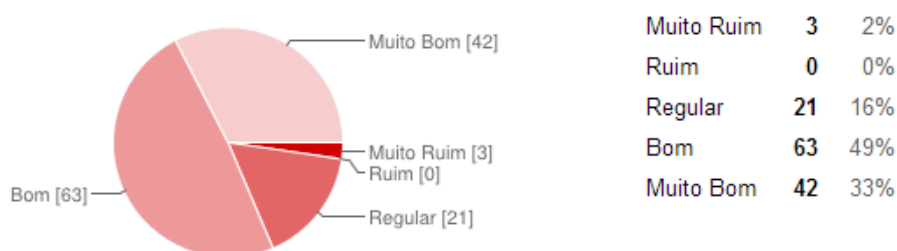


#### 5.4.4 Avaliação do ViajeTriFácil

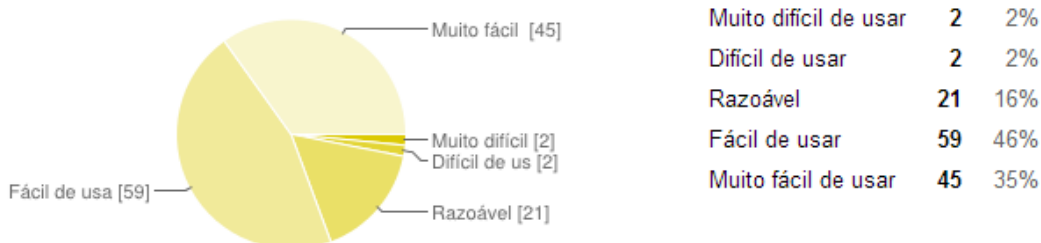
As questões relativas ao ViajeTriFácil avaliaram a opinião do usuário sobre a aparência da interface em termos de qualidade dos gráficos, das cores e dos textos; a facilidade de uso do sistema; as funcionalidades; o tempo de resposta; e a qualidade das informações apresentadas.

De acordo com as respostas ilustradas na Figura 5.15, a aparência da interface gráfica é agradável a grande parte dos usuários, sendo um sistema de fácil utilização e que atende boa parte das necessidades dos usuários de transporte público.

##### Qual sua opinião sobre a aparência da interface gráfica do sistema proposto?



##### E o que você acha da facilidade de uso do sistema?



##### As funcionalidades existentes atendem as suas necessidades como usuário?

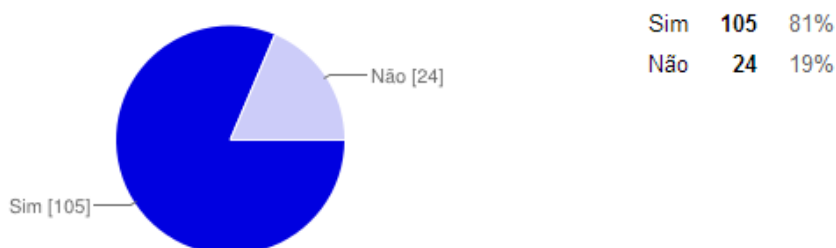
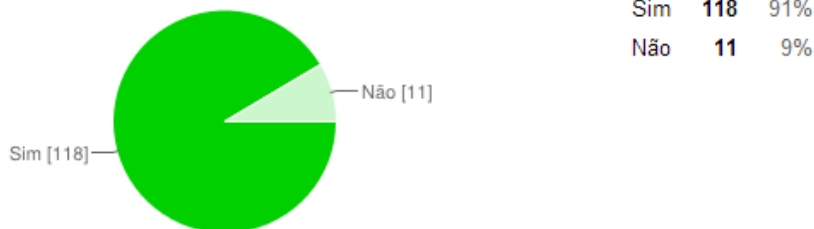


Figura 5.15: Pesquisa de opinião – avaliação da aparência, facilidade de uso e funcionalidades.

As respostas apresentadas na Figura 5.16 mostram que a ferramenta dá boas respostas para os usuários, fornecendo informações reais, úteis e relevantes para a escolha do trajeto, tendo um tempo de resposta satisfatório.

#### O tempo de resposta de uma pesquisa é satisfatório?



#### As opções obtidas como resposta fornecem informações úteis para a escolha do trajeto?

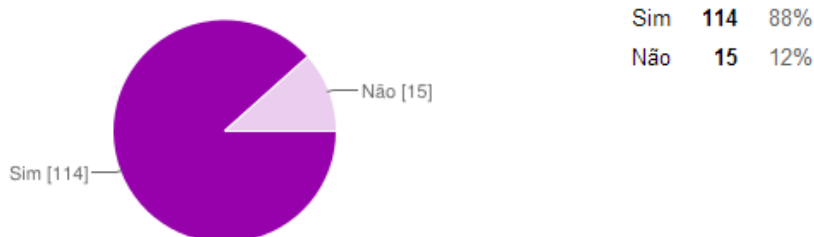


Figura 5.16: Pesquisa de opinião – avaliação do tempo de resposta e da qualidade das informações apresentadas.

### 5.4.5 Opiniões sobre o ViajeTriFácil

Além das questões objetivas aplicadas para avaliar o ViajeTriFácil, foi incluída uma última questão descritiva, opcional, em que o usuário poderia relatar problemas, deixar sugestões de melhorias, novas funcionalidades ou mesmo algum comentário adicional. Essa questão foi respondida por 61 das 129 pessoas que realizaram a avaliação.

#### 5.4.5.1 Problemas relatados

Em linhas gerais, os principais problemas relatados foram:

- Em algumas marcações de pontos no modo gráfico, as caixas de texto dos endereços de origem e destino são preenchidas com informações incompletas. Muitas vezes o sistema apresenta apenas o município e o CEP do local selecionado;
- Em algumas consultas no modo texto, quando selecionado o endereço de origem ou destino nas opções exibidas pelo recurso de autocompletar, a caixa de endereço não apresenta a mesma informação da opção selecionada;
- Não está claro o procedimento para selecionar os pontos de origem e destino clicando diretamente no mapa;
- Exibir apenas três alternativas de transporte para cada pesquisa realizada;
- Sugerir opções que nem sempre são as melhores, não levando em conta que se o usuário caminhar um pouco mais pode utilizar apenas um ônibus, ou

realizar um número menor de integrações. Exemplo: o trajeto do Campus do Vale da UFRGS até a Av. João Pessoa não sugere a linha 343 como opção;

- Exibir múltiplas opções em cima de um mesmo ônibus, variando apenas o horário de partida;
- Exibir informações incorretas sobre ruas e locais no mapa;
- Apresentar termos em inglês na descrição dos trechos a percorrer.

Para alguns problemas relatados, como tornar mais claro o procedimento para selecionar origem e destino no mapa, e a exibição de apenas três opções, a solução já existe e não há grandes problemas para implementá-la, mas, mesmo assim, dependerão de uma avaliação prévia para garantir o bom funcionamento do sistema. Contudo, para os demais problemas relatados, a solução não é simples, pois envolve a alteração de outros sistemas como o OSM e o OTP.

Para a exibição correta dos endereços de origem e destino, por exemplo, é necessário o estudo das opções adicionais no uso da biblioteca *Google Maps Javascript API v3*, de modo a analisar a viabilidade de apresentar os endereços de forma mais amigável.

A exibição de opções de trajeto apresentadas pela ferramenta, que nem sempre são as melhores, pode ser melhorada sim, até porque a melhoria é constante, mas deve passar por um estudo para averiguar, caso a caso, qual o real motivo do problema e qual a solução que deve ser tomada. Para alguns casos, como trajetos a pé sugeridos incorretamente, a atualização das informações cartográficas no *servidor OSM* deve solucionar o problema. Para outros, a correção de informações de transporte público como a inclusão de paradas e de outros modais na especificação GTFS devem ser a solução. E, caso essas mudanças não resolvam, ainda existem outras opções como a alteração no algoritmo de roteamento.

A exibição de múltiplas opções com o mesmo ônibus ocorre porque o algoritmo de roteamento do OTP não diferencia as melhores opções encontradas pelo ônibus utilizado. Portanto, se a próxima opção encontrada pelo algoritmo utilizar o mesmo ônibus, somente com o horário diferente, ela será considerada. Para resolver esse problema o algoritmo deve ser modificado.

A correção nas informações sobre ruas e locais depende da comunidade OSM, pois são essas pessoas que alimentam o *servidor OSM* com os dados necessários para formar o mapa.

A correção de termos em inglês, como “*corner of*” e “*path*”, duas ações são necessárias: atualização dos endereços no *servidor OSM* e alteração do idioma do OTP para português.

#### 5.4.5.2 Sugestões de melhorias

Além de problemas, alguns usuários deram sugestões de mudanças e novas funcionalidades para melhorar a ferramenta. Dentre elas, as principais foram:

- Sugerir, no recurso de autocompletar dos endereços de origem e destino, apenas opções com endereços de Porto Alegre;
- Permitir inverter os endereços de origem e destino;

- Pesquisar as opções logo após o usuário preencher os pontos de origem e destino, sem ser necessário pressionar o botão “*Planejar viagem*”;
- Permitir ao usuário limitar a quantidade de linhas de ônibus por viagem;
- Exibir as paradas mais próximas do ponto de origem para que o usuário possa escolher qual utilizar e também exibir, com maior clareza, as paradas que o usuário deve descer;
- Dar maior destaque a informação de quantos ônibus são necessários para percorrer o trajeto;
- Exibir, no mapa, qual o sentido de cada linha de ônibus;
- Exibir informações sobre as linhas de ônibus em tempo real;
- Exibir as opções de trajetos, no painel da esquerda, em cores diferentes;
- Possibilitar o acesso à tabela de horário dos ônibus;
- Sinalizar linhas que não operam diariamente;
- Permitir a inserção de linhas de forma colaborativa, permitindo que os próprios usuários cadastrem dados de ônibus pertencentes a outras cidades;
- Incluir os modais de transporte bicicleta, lotação e trem;
- Salvar preferências de busca do usuário;
- Disponibilizar aplicativo para iOS;
- Exibir os mapas na versão para dispositivos móveis;
- Obter localização por GPS na versão para dispositivos móveis.

## 5.5 Considerações finais

Neste capítulo foram apresentadas as validações realizadas sobre o ViajeTriFácil, as quais foram feitas através de testes funcionais, onde foram analisados alguns cenários sobre a funcionalidade principal da ferramenta – o planejamento de viagem - para avaliar sua corretude com base no conhecimento prévio de trajetos específicos; e pela aplicação de um questionário avaliado por 129 pessoas em um período específico com o intuito de averiguar o perfil do usuário, a usabilidade, a corretude, o desempenho do sistema e também coletar informações sobre problemas encontrados e sugestões de melhoria.

Com a sequência de testes aplicados é possível concluir que a ferramenta funciona e cumpre com o seu objetivo principal, que é suprir a falta de informações sobre trajetos e itinerários de ônibus em Porto Alegre através de uma ferramenta que permite a pesquisa desses dados.

Através dos inúmeros comentários feitos por usuários também ficou claro que existem problemas que ainda precisam ser solucionados, e que diversas melhorias podem ser feitas para tornar a ferramenta mais funcional e oferecer mais possibilidades aos seus usuários. A iniciativa e a solução inicial já foram feitas neste trabalho, as melhorias são contínuas.

Para concluir a avaliação, a Tabela 5.1, com base na Tabela 2.3, apresenta novamente as principais funcionalidades existentes nas soluções de TIC analisadas na seção 2.5 e inclui o ViajeTriFácil, apresentando o que ele oferece em relação às demais soluções.

Tabela 5.1: Comparativo entre as principais funcionalidades de algumas soluções de TIS (*Traveler Information Systems*) existentes

	<i>Poa Bus</i>	<i>Poa Transporte</i>	<i>Google Transit</i>	<i>Moovit</i>	<i>Nosso Bus</i>	<i>Antares</i>	<i>Moobly</i>	<i>Mapa Interativo de Buenos Aires</i>	<i>Viaje Tri Fácil</i>
Gratuito	✓	✓	✓	✓	✓	✓	✓	✓	✓
Baseado em Software Livre	✗	✗	✗	✗	✗	✓	✗	✓	✓
Pesquisa de linhas	✓	✓	✗	✗	✗	✓	✓	✗	✗
Pesquisa textual de rotas	✓	✗	✓	✓	✓	✓	✓	✓	✓
Pesquisa gráfica (mapa) de rotas	✓	✗	✓	✓	✗	✓	✓	✓	✓
Integrações (baldeações)	✗	✗	✓	✓	✗	✓	✗	✓	✓
Múltiplos modais de transporte	✗	✓	✓	✓	✗	✗	✗	✓	✗
Hora de partida e chegada	✗	✗	✓	✓	✗	✓	✗	✗	✓
Tempo de viagem	✗	✗	✓	✓	✗	✓	✗	✓	✓
Versão web	✓	✓	✓	✓	✗	✓	✗	✓	✓
Versão web responsiva	✗	✗	✓	✗	✗	✗	✗	✗	✓
Versão para dispositivos móveis	✗	✗	✓	✓	✓	✗	✓	✓	✓

## 6 CONCLUSÃO

Ao longo deste trabalho foram apresentadas as etapas realizadas na construção do ViajeTriFácil. Iniciou-se pela discussão da importância do transporte no desenvolvimento de um país e dos problemas gerados pela priorização do transporte individual e pelo crescimento acelerado e mal planejado dos centros urbanos. Foram discutidas soluções já empregadas em várias partes do mundo, as quais convergem para o emprego do transporte coletivo em detrimento do individual, destacando-se a obtenção de informações de maneira fácil, rápida e precisa como forma de incentivar o uso do transporte coletivo.

O objetivo deste trabalho é de elaborar um sistema baseado em software livre para consultar trajetos de ônibus aplicado a cidade de Porto Alegre. Para atingir esse objetivo foram estudados os principais conceitos de Sistemas de Transporte Inteligente (ITS). Também foram apresentados os sistemas *OpenStreetMap* (OSM) e *OpenTripPlanner* (OTP), ferramentas colaborativas construídas sobre licenças abertas, as quais foram necessárias para permitir a obtenção e interação do usuário com o mapa e para calcular a rota a percorrer pelo usuário, respectivamente. Por fim, ainda foram apresentados alguns sistemas existentes com o intuito de mostrar que soluções como a que está sendo proposta neste trabalho não é única.

De posse desse estudo, a solução foi especificada, projetada, construída e avaliada. Na especificação foram feitas a análise de requisitos, o diagrama de casos de uso e a prototipação de interfaces. No projeto foram definidos quais os padrões arquiteturais mínimos que o sistema deveria seguir. Com posse dessas informações, o sistema foi implementado utilizando diversas tecnologias, sendo habilitado para uso tanto em navegadores web quanto em dispositivos móveis. E, então, o sistema foi avaliado através de testes funcionais aplicados com base no conhecimento prévio de alguns trajetos e também por um questionário disponibilizado na Internet para avaliar a opinião dos usuários sobre a ferramenta.

As avaliações realizadas sobre a ferramenta mostraram que ela funciona e cumpre com seu objetivo principal, que é fornecer informações sobre trajetos, itinerários e horários de ônibus na cidade de Porto Alegre, o qual foi destacado na introdução deste trabalho como um dos principais fatores que motivam o emprego do transporte coletivo, e também que ela apresentou uma boa aceitação pelos usuários, tendo avaliações positivas sobre sua usabilidade, corretude e desempenho, além de contar com inúmeros comentários sugerindo correções e melhorias.

Com a realização deste trabalho ficou evidente que ele é uma ótima ferramenta, trazendo informações úteis para seus usuários, e tendo grande potencial de crescimento. O grande diferencial do ViajeTriFácil em relação a maioria das ferramentas existentes está na tecnologia empregada para sua construção, a qual utiliza, em grande parte,

soluções em software livre como o *OpenStreetMap* (OSM), o *OpenTripPlanner* (OTP), o *Bootstrap* e o *PhoneGap*, com exceção da busca textual, a qual emprega a solução do Google que, mesmo sendo gratuita, não é software livre. Além disso, a ferramenta também tem alguns pontos positivos em relação a algumas soluções, pois provê informações de trajetos a pé e integrações (baldeações), pode ser utilizada a partir de qualquer navegador web e grande parte dos dispositivos móveis, e pode ser facilmente adaptada para cidades de qualquer tamanho.

Como trabalhos futuros podem ser analisadas e implementadas as sugestões de melhorias apontadas pelos usuários no questionário de avaliação. Ainda, tornar a ferramenta totalmente baseada em software livre, utilizando o *Nominatim* como ferramenta de busca textual. Para isso é necessário um investimento no desenvolvimento do próprio *Nominatim* e nas informações disponíveis pela comunidade OSM. Outra sugestão é criar soluções nativas para dispositivos móveis, visto que as soluções utilizadas são apenas uma adaptação do sistema web feita através do *PhoneGap*. Mais uma opção que seria útil acrescentar à ferramenta seria a criação de um editor gráfico, disponível na web, para especificações GTFS. Esse editor possibilitaria que colaboradores ou mesmo órgãos públicos pudessem expandir a região compreendida pelo ViajeTriFácil.

## REFERÊNCIAS

- AJAX W3SCHOOLS. **AJAX Tutorial**. [S.l.:s.n], 2014. Disponível em: <<http://www.w3schools.com/ajax/default.asp>>. Acesso em: jun. 2014.
- ATRICE. **PhoneGap Explained Visually**. [S.l.:s.n], 2012. Disponível em: <<http://phonegap.com/2012/05/02/phonegap-explained-visually>>. Acesso em: jun. 2014.
- BASS, Len; Clements, Paul; Kazman, Rick. **Software Architecture in Practice, 2<sup>nd</sup> edition**. [S.l.], Addison-Wesley, 2003.
- BAZZAN, A. L. C.; Klügl, F. **Introduction to Intelligent Systems in Traffic and Transportation**. Morgan & Claypool Publishers, 2014.
- CROCKFORD, Douglas. **JavaScript: The Good Parts**. Sebastopol, CA, USA. O'Reilly Media Inc., 2008.
- CSS3 W3C. **Introduction to CSS3**. [S.l.:s.n], 2001. Disponível em: <<http://www.w3.org/TR/2001/WD-css3-roadmap-20010523>>. Acesso em: jun. 2014.
- DELLING, Daniel; PAJOR, Thomas; WERNECK, Renato. **Round-Based Public Transit routing**. 2012. 11 f. Proceedings of the 14th Meeting on Algorithm Engineering and Experiments – Society for Industrial and Applied Mathematics.
- DENATRAN. **Frota de veículos**. [S.l.:s.n], 2014. Disponível em: <<http://www.denatran.gov.br/frota.htm>>. Acesso em: fev. 2014.
- DEMOS OTP. **OpenTripPlanner - Demos**. [S.l.:s.n], 2014. Disponível em: <<https://github.com/opentripplanner/OpenTripPlanner/wiki/Demos>>. Acesso em: abr. 2014.
- DIJKSTRA, Edsger. A Note on Two Problems in Connexion with Graphs. **Numerische Mathematik**, Amsterdam, v.1, n.1, p. 269-271, jun. 1959.
- DISCOVERY. **Soluções para o Trânsito – São Paulo**. [S.l.:s.n], 2012. Disponível em <<http://www.youtube.com/watch?v=dGSdnIOuZvk>>. Acesso em: fev. 2014.
- ECMA. **The JSON Data Interchange Format**. Genebra, 1<sup>st</sup> ed, out. 2013. Disponível em <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>. Acesso em: abr. 2014.



- ESRI. **ESRI Shapefile Technical Description**. [S.l.:s.n], mar. 1998. Disponível em <<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>>. Acesso em: abr. 2014.
- FABICO. **Porto Alegre em Pauta – Mobilidade Urbana Bloco 1**. [S.l.:s.n], 2012-a. Disponível em <<http://www.youtube.com/watch?v=4AWtlQIwyU>>. Acesso em: fev. 2014.
- FABICO. **Porto Alegre em Pauta – Mobilidade Urbana Bloco 2**. [S.l.:s.n], 2012-b. Disponível em <<http://www.youtube.com/watch?v=Qg7LRxVChqY>>. Acesso em: fev. 2014.
- FAQ OTP. **Frequently Asked Questions**. [S.l.:s.n], 2013. Disponível em: <<https://github.com/opentripplanner/OpenTripPlanner/wiki/FrequentlyAskedQuestions>>. Acesso em: abr. 2014.
- FIELDING, Roy T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. 162 f. Dissertação (Doutorado em Ciência da Computação) – University of California, Irvine.
- FILHO, Wilson de P. P. **Engenharia de Software – Fundamentos, Métodos e Padrões**. LTC, 2009.
- GEISBERGER R. **Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks**. 2008. 70 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Tecnologia de Karlsruhe, Karlsruhe, Alemanha.
- GPL. **GNU General Public License**. [S.l.]: Free Software Foundation, 2007. Disponível em: <<https://www.gnu.org/copyleft/gpl.html>>. Acesso em: abr. 2014.
- GRAPH OTP. **OpenTripPlanner - GraphBuilder**. [S.l.:s.n], 2014. Disponível em: <<https://github.com/opentripplanner/OpenTripPlanner/wiki/GraphBuilder>>. Acesso em: abr. 2014.
- GOOGLE TRANSIT. **GTFS – Overview**. [S.l.:s.n], jan. 2012. Disponível em: <<https://developers.google.com/transit/gtfs/>>. Acesso em: abr. 2014.
- GUEDES, Gilleanes T. A. **UML – Uma Abordagem Prática**. São Paulo, SP, Brasil: Novatec Editora Ltda, 2004.
- HART, P. E., NILSSON, N. J., RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. **Systems Science and Cybernetics**, IEEE Transactions, v.4, n.2, p. 100-107, jul. 1968.
- HTML5 W3SCHOOLS. **HTML5 - Introduction**. [S.l.:s.n], 2014. Disponível em: <[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)>. Acesso em: jun. 2014.
- IPEA. **Indicadores de mobilidade urbana da PNAD 2012**. [S.l.:s.n], 2013. 48 f. Disponível em: <[http://www.ipea.gov.br/portal/images/stories/PDFs/comunicado/131024\\_comunicadoipea161.pdf](http://www.ipea.gov.br/portal/images/stories/PDFs/comunicado/131024_comunicadoipea161.pdf)>. Acesso em: fev. 2014.
- ID. **OpenStreetMap - iD**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/iD>>. Acesso em: abr. 2014.

JAIQUES, P. et al. **Um Estudo de Caso com o Uso do A\* para o Problema do Caminho Mais Curto em Transporte Público**. 2012. 10 f. Encontro Nacional de Inteligência Artificial, Curitiba.

JETTY. **Eclipse - Jetty**. [S.l.]: The Eclipse Foundation, 2014. Disponível em: <<http://www.eclipse.org/jetty>>. Acesso em: abr. 2014.

JOSM. **OpenStreetMap - JOSM**. [S.l.:s.n], 2014. Disponível em: <<https://josm.openstreetmap.de/>>. Acesso em: abr. 2014.

LGPL. **GNU Lesser General Public License**. [S.l.]: Free Software Foundation, 2007. Disponível em: <<https://www.gnu.org/licenses/lgpl.html>>. Acesso em: abr. 2014.

MAPNIK. **Mapnik**. [S.l.:s.n], 2013. Disponível em: <<http://mapnik.org>>. Acesso em: abr. 2014.

MENDONÇA, Cláudio. **Transporte Rodoviário: Por que o Brasil depende tanto desse sistema**. [S.l.:s.n], 2006. Disponível em: <<http://educacao.uol.com.br/disciplinas/geografia/transporte-rodoviario-por-que-o-brasil-depende-tanto-desse-sistema.htm>>. Acesso em: fev. 2014.

NAVIT. **Navit**. [S.l.:s.n], 2014. Disponível em: <<http://www.navit-project.org>>. Acesso em: abr. 2014.

NIELSEN, Jakob. **Response Times: The 3 Important Limits**. [S.l.:s.n], 1993. Disponível em <<http://www.nngroup.com/articles/response-times-3-important-limits>>. Acesso em: jun. 2014.

NOMINATIM. **Nominatim**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/Nominatim>>. Acesso em: abr. 2014.

OSM2PGSQL. **OpenStreetMap – Osm2pgsql**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/Osm2pgsql>>. Acesso em: abr. 2014.

OSM3S. **OpenStreetMap – Overpass API**. [S.l.:s.n], 2014. Disponível em: <[http://wiki.openstreetmap.org/wiki/Overpass\\_API](http://wiki.openstreetmap.org/wiki/Overpass_API)>. Acesso em: abr. 2014.

OSMAPI. **OpenStreetMap – API**. [S.l.:s.n], 2013. Disponível em: <<http://wiki.openstreetmap.org/wiki/API>>. Acesso em: abr. 2014.

OSMCONVERT. **OpenStreetMap – Osmconvert**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/Osmconvert>>. Acesso em: abr. 2014.

OSMOSIS. **OpenStreetMap – Osmosis**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/Osmosis>>. Acesso em: abr. 2014.

OTP. **OpenTripPlanner - Home**. [S.l.:s.n], 2014. Disponível em: <<https://github.com/opentripplanner/OpenTripPlanner/wiki>>. Acesso em: abr. 2014.

OTP OSM. **OpenTripPlanner**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/OpenTripPlanner>>. Acesso em: abr. 2014.

PASIN, Márcia. Suporte Computacional para a Implantação de Redes de Transporte para Cidades Inteligentes. **ERAD/RS 2014**, Alegre, p. 39-56, mar. 2014.

- PBFTOOSM. **OpenStreetMap – Pbftoosm**. [S.l.:s.n], 2011. Disponível em: <<http://wiki.openstreetmap.org/wiki/Pbftoosm>>. Acesso em: abr. 2014.
- PENCIL. **Pencil Project**. [S.l.:s.n], 2012. Disponível em: <<http://pencil.evolus.vn>>. Acesso em: mai. 2014.
- POSTGIS. **PostGIS**. [S.l.:s.n], 2014. Disponível em: <<http://postgis.net/>>. Acesso em: abr. 2014.
- POTLATCH. **OpenStreetMap - Potlatch**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/Potlatch>>. Acesso em: abr. 2014.
- PRESSMAN, Roger. **Software Engineering – A Practitioner’s Approach, 7<sup>th</sup> edition**. New York, NY, USA: McGraw-Hill, 2010.
- RAPTOR OTP. **OpenTripPlanner - Raptor**. [S.l.:s.n], 2013. Disponível em: <<https://github.com/opentripplanner/OpenTripPlanner/wiki/Raptor>>. Acesso em: abr. 2014.
- REST WIKIPEDIA. **Representational state transfer**. [S.l.:s.n], 2014. Disponível em: <[http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)>. Acesso em: abr. 2014.
- ROUTING OTP. **OpenTripPlanner - RoutingBibliography**. [S.l.:s.n], 2014. Disponível em: <<https://github.com/opentripplanner/OpenTripPlanner/wiki/RoutingBibliography>>. Acesso em: abr. 2014.
- SILVA, Danyela Moraes da. **Sistemas Inteligentes no Transporte Público Coletivo por Ônibus**. 2000. 144 f. Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia, UFRGS, Porto Alegre.
- SLIPPYMAP. **Slippy Map**. [S.l.:s.n], 2013. Disponível em: <[http://wiki.openstreetmap.org/wiki/Slippy\\_Map](http://wiki.openstreetmap.org/wiki/Slippy_Map)>. Acesso em: abr. 2014.
- SOMMERVILLE, Ian; SAWYER, Pete. **Requirements Engineering: A Good Practice Guide, 1<sup>st</sup> edition**. New York, NY, USA: John Wiley & Sons, 1997.
- SUSSMANN, Joseph. **Introduction to Transportation Systems**. Boston, London: Artech House, 2000
- SWITCH2OSM. **Switch2Osm – Manually building a tile server**. [S.l.:s.n], 2013. Disponível em: <<http://switch2osm.org/serving-tiles/manually-building-a-tile-server/>>. Acesso em: abr. 2014.
- TANEMBAUM, Andrew S.; Steen Maarten Van. **Sistemas Distribuídos : princípios e paradigmas, 2. ed**. São Paulo : Pearson Prentice Hall, 2007.
- TECMUNDO. **O que é API?**. [S.l.:s.n], 2009. Disponível em: <<http://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>>. Acesso em: abr. 2014.
- TOMCAT. **Apache Tomcat**. [S.l.]: The Apache Software Foundation, 2014. Disponível em: <<http://tomcat.apache.org>>. Acesso em: abr. 2014.

TRAC. **GeoTIFF**. [S.l.:s.n], 2009. Disponível em: <<http://trac.osgeo.org/geotiff/>>. Acesso em: abr. 2014.

TRANSFETY. **Road Engineering Journal: Study Compares Older and Younger Pedestrian Walking Speeds**. [S.l.:s.n], 1997. Disponível em <<http://www.usroads.com/journals/p/rej/9710/re971001.htm>>. Acesso em: jun. 2014.

U.S. DOT. **Advanced Public Transportation Systems: The State of the Art – Update 2000**. [S.l.:s.n], 2000. Disponível em <[http://ntl.bts.gov/lib/jpodocs/repts\\_te/13583.pdf](http://ntl.bts.gov/lib/jpodocs/repts_te/13583.pdf)>. Acesso em: mar. 2014.

U.S. DOT. **Advanced Rural Transportation Systems (ARTS) – Strategic Plan**. [S.l.:s.n], 1997. Disponível em <<http://ntl.bts.gov/lib/16000/16000/16024/PB2000102164.pdf>>. Acesso em: mar. 2014.

WIEGERS, Karl E. **Software Requirements, 2nd Edition**. Redmond. Washington: Microsoft Press, 2003.

WGS WIKIPEDIA. **World Geodetic System**. [S.l.:s.n], 2014. Disponível em: <[http://en.wikipedia.org/wiki/World\\_Geodetic\\_System](http://en.wikipedia.org/wiki/World_Geodetic_System)>. Acesso em: abr. 2014.

XAPI. **OpenStreetMap – XAPI**. [S.l.:s.n], 2014. Disponível em: <<http://wiki.openstreetmap.org/wiki/Xapi>>. Acesso em: abr. 2014.

## APÊNDICE

### QUESTIONÁRIO PARA AVALIAR UM SISTEMA DE CONSULTAS DE TRAJETOS DE ÔNIBUS

Este questionário tem por objetivo coletar dados para avaliar, do ponto de vista do usuário, o perfil e a usabilidade de um sistema desenvolvido através do uso de tecnologias em software livre para a consulta de trajetos de ônibus existentes entre dois pontos distintos na cidade de Porto Alegre.

O formulário foi gerado através do Google Forms e está disponível em <https://docs.google.com/forms/d/1V0QZksEvPBPY4xQ8-3MtKMoWIN-8DA3rnCWuR4YDZwE/viewform>. Através desse recurso é possível coletar as respostas dos usuários em uma tabela e gerar gráficos a partir delas.

#### **Perfil do usuário**

1. Sexo?
  - Masculino
  - Feminino
2. Qual sua faixa etária?
  - Até 18 anos
  - De 19 a 30 anos
  - De 31 a 45 anos
  - De 46 a 60 anos
  - De 61 a 75 anos
  - Mais de 75 anos
3. Qual seu nível de escolaridade?
  - Ensino Fundamental
  - Ensino Médio
  - Ensino Superior
  - Pós-Graduação

#### **Mobilidade**

4. Quantas vezes por semana você usa ônibus em seu deslocamento?
  - Nunca
  - 1
  - 2
  - 3

- 4
  - 5
  - 6
  - Todos os dias
5. Na sua opinião, qual ou quais são os problemas existentes com o transporte público na cidade de Porto Alegre?
- Falta de informação sobre trajetos e itinerários
  - Superlotação
  - Congestionamento
  - Pontualidade
  - Inexistência de linhas para destinos desejados
  - Conforto, limpeza e conservação da frota de ônibus
  - Outros

### Softwares de Mobilidade Urbana

6. Você acha que a Internet e as redes sociais podem auxiliar os usuários na consulta sobre trajetos e itinerários?
- Sim
  - Não
7. Você já utilizou algum sistema (web ou mobile) para encontrar meios de transporte público (metrô, ônibus, trem ou outros)?
- Sim
  - Não
8. Se a resposta da questão anterior for sim, então qual ou quais são?
- BA Cómo Llego (<http://mapa.buenosaires.gob.ar>)
  - EPTC (<http://www2.portoalegre.rs.gov.br/eptc>)
  - GoogleTransit (<https://maps.google.com/transit>)
  - Moobly (<http://www.moobly.com.br>)
  - Moovit (<http://www.moovitapp.com>)
  - PoaBus (<http://www.poabus.com.br>)
  - PoaTransporte (<http://www.poatransporte.com.br>)
  - Poa Trânsito (<http://www.poaemtransito.com.br>)
  - Outros
9. Qual sua opinião sobre a aparência da interface gráfica do sistema proposto?
- Muito Ruim
  - Ruim
  - Regular
  - Bom
  - Muito Bom
10. E o que você acha da facilidade de uso do sistema?
- Muito difícil de usar
  - Difícil de usar
  - Razoável
  - Fácil de usar

- Muito fácil de usar
11. O tempo de resposta de uma pesquisa é satisfatório?
- Sim
  - Não
12. As opções obtidas como resposta fornecem informações úteis para a escolha do trajeto?
- Sim
  - Não
13. As funcionalidades existentes atendem as suas necessidades como usuário?
- Sim
  - Não
14. Caso tenha alguma sugestão de nova funcionalidade, melhoria ou algo mais deixe seu comentário aqui: