



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

ESCOLA DE INFORMÁTICA APLICADA

Sistema de Apoio à Mobilidade Urbana com Dados Abertos

Uriel dos Santos Pereira Bertoche

**Orientador**

Márcio de Oliveira Barros

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 2016

Sistema de Apoio à Mobilidade Urbana com Dados Abertos.

Uriel dos Santos Pereira Bertoche

Projeto de Graduação apresentado à Escola de  
Informática Aplicada da Universidade Federal do  
Estado do Rio de Janeiro (UNIRIO) para obtenção do  
título de Bacharel em Sistemas de Informação.

Aprovada por:

---

Prof. Márcio de Oliveira Barros, DSc. (UNIRIO)

---

Profª. Morganna Carmem Diniz, DSc. (UNIRIO)

---

Profª. Fernanda Araujo Baião, DSc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

DEZEMBRO DE 2016

## **Agradecimentos**

## **RESUMO**

A mobilidade urbana nas grandes cidades é um desafio a ser enfrentado pelas entidades públicas e pela própria população. Uma medida paliativa para os efeitos causados pelas deficiências do transporte público é o uso de aplicativos para contornar estes problemas a partir de decisões informadas sobre as opções de mobilidade.

O objetivo deste projeto é desenvolver um serviço de descoberta de rotas para o deslocamento entre dois pontos dentro do município do Rio de Janeiro, utilizando dados abertos disponíveis no Portal de Dados Abertos da Prefeitura da Cidade do Rio de Janeiro, com o intuito de contribuir com a melhoria da mobilidade dos usuários.

Comparamos os resultados gerados pelo serviço desenvolvidos no escopo deste projeto com aplicativos semelhantes para verificar a eficácia e precisão das informações geradas por nossos algoritmos. Os resultados iniciais são positivos, mostrando que o serviço gera rotas compatíveis com os melhores aplicativos disponíveis.

**Palavras-chave:** Mobilidade urbana, dados abertos, transporte público, aplicativos.

## **ABSTRACT**

Urban mobility in big cities is a challenge to be faced by public entities and the population itself. Mobile applications are often used as palliative measures used to dampen the effects caused by the deficiencies of present day public transportations, allowing their users to make informed decisions on which transportation services would be better suited for them.

The objective of the present project is to offer a route discovery service for the transportation between two points in Rio de Janeiro city, using the open data available at "Portal de Dados Abertos da Prefeitura da Cidade do Rio de Janeiro", in order to improve the user's mobility.

We compared the results generated by the service developed in the scope of this project with similar applications to verify the effectiveness and precision of the information generated by our algorithms. Initial results are promising, as the service generates routes compatible with the best available applications.

**Keywords:** Urban mobility, open data, public transport, apps.

## **Índice**

<b>Introdução</b>	7
Motivação	7
Objetivos	7
Organização do texto	7
<b>Mobilidade Urbana</b>	9
Mobilidade urbana	9
Dados abertos	10
Dados abertos na Cidade do Rio de Janeiro	12
Aplicativos de mobilidade urbana	12
Considerações finais	15
<b>Solução Proposta</b>	16
Um histórico da proposta	16
A Implementação da proposta final	18
O problema dos dados disponibilizados	21
<b>Exemplo de Uso</b>	26
Projeto Exemplo	26
Exemplo de Uso do Sistema de Busca de Comutações	28
Considerações Finais	31
<b>Conclusão</b>	32
Contribuições	32
Trabalhos futuros	32
Limitações do estudo	33

# **1 Introdução**

## **1.1 Motivação**

A falta de confiabilidade e previsibilidade no transporte público do Rio de Janeiro é um problema que afeta os cidadãos que utilizam esse serviço diariamente. Apesar do Rio de Janeiro ser uma cidade grande e turística, ou talvez justamente por ser uma cidade grande onde o meio de transporte público mais eficaz são os ônibus, é difícil encontrar informações sobre qual ou quais ônibus podemos pegar para chegar a um determinado lugar.

Com esses dois problemas em mente, tentamos criar um sistema que pudesse auxiliar o cidadão carioca e turistas visitando a cidade a minimizar seus problemas de mobilidade urbana.

## **1.2 Objetivos**

Inicialmente, o objetivo deste projeto era criar um sistema que utilizaria os dados dos ônibus municipais do Rio de Janeiro, disponibilizados pelo Portal de dados abertos da Prefeitura do Rio, para auxiliar os usuários a encontrar as linhas de ônibus que os levassem ao destino desejado e informar o tempo estimado de chegada dos ônibus dessas linhas no ponto de parada que seria usado pelo usuário.

Devido a problemas técnicos (que serão detalhados adiante neste documento), o objetivo inicial precisou ser revisto e reduzido a apenas resolver o problema dos itinerários. Dado um ponto de partida e um ponto de destino desejado, o sistema informará ao usuário quais as linhas que ele pode usar e, se for preciso fazer uma baldeação, onde ela deve ser feita.

## **1.3 Organização do texto**

O presente trabalho está estruturado em capítulos e será desenvolvido da seguinte forma:

- Capítulo II: Mobilidade urbana – Abordagem dos conceitos e análise de outros sistemas e aplicativos para dispositivos móveis com propósito semelhante;
- Capítulo III: Solução proposta – Histórico do que foi feito, dos problemas encontrados durante o desenvolvimento da solução inicial, mudanças no projeto e resultado final;
- Capítulo IV: Exemplos de uso – Montagem com possíveis cenários de uso do sistema, fazendo comparação do resultado sugerido com outras opções de aplicações;
- Capítulo V: Conclusão – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.



## **2 Mobilidade Urbana**

Neste capítulo falaremos sobre a mobilidade urbana e o problema existente devido ao grande número de carros em circulação nas áreas urbanas das grandes cidades. Falaremos também a respeito dos dados abertos, explicando o conceito, suas leis e princípios. Finalmente, faremos um estudo de aplicativos na área de mobilidade urbana a fim de compreender melhor as opções existentes hoje para o cidadão, seus pontos fortes e fracos e acurácia das informações.

### **2.1 Mobilidade urbana**

Mobilidade Urbana é a condição em que se realizam os deslocamentos de pessoas e cargas no espaço urbano de um Município. Assim, a mobilidade urbana adequada é obtida por meio de políticas de transporte e circulação que visam a melhoria da acessibilidade e mobilidade das pessoas e cargas no espaço urbano, através da priorização dos modos de transporte coletivo e não motorizados de maneira efetiva, socialmente inclusiva e ecologicamente sustentável [2].

A mobilidade urbana é hoje um desafio das grandes cidades. Refere-se ao trânsito de pessoas e cargas dentro da área urbana da cidade, e com o aumento da população urbana e da demanda gerada pelo crescente mercado, se não tratada de maneira adequada, pode se tornar um sério problema.

Na região Sudeste, em 2010, cerca de 50% da população utilizava o transporte público como principal meio de transporte e cerca de 25% utilizava o carro [1]. Considerando que a quantidade de meios de transporte público necessária para movimentar um mesmo número de pessoas é muito menor que a quantidade de carros, podemos perceber que a maior parte do trânsito urbano é composto de carros. A Tabela 1 apresenta a distribuição de uso dos diversos modais urbanos entre as regiões do país no ano de 2010.

TABELA 1 - Distribuição de uso dos modais de transporte urbano

	<b>Brasil</b>	<b>Sul</b>	<b>Sudeste</b>	<b>Centro-Oeste</b>	<b>Nordeste</b>	<b>Norte</b>
<b>Transporte público</b>	44,3	46,3	50,7	39,6	37,5	40,3
<b>Carro</b>	23,8	31,7	25,6	36,5	13,0	17,6
<b>Moto</b>	12,6	12,4	11,6	6,5	19,4	8,2
<b>À pé</b>	12,3	7,6	8,3	13,7	18,8	16,1
<b>Bicicleta</b>	7,0	2,0	3,8	3,7	11,3	17,9

Fonte: SIPS/IPEA, 2010

A grande quantidade de veículos nas ruas gera um número de problemas: quanto mais carros, mais trânsito, quanto mais trânsito maior o consumo de combustível e quanto maior o consumo de combustível, maior a emissão de poluentes. Sem contar os problemas de saúde relacionados a essa poluição e a redução na qualidade de vida dos cidadãos [11], que podem repercutir de maneiras mais sutis no dia a dia da cidade.

Dentre as razões para o grande número de carros nas cidades, podemos destacar a má qualidade do serviço público, muitas vezes não atendendo às necessidades dos cidadãos ou não sendo eficiente, o aumento da renda média dos brasileiros nos últimos anos e os incentivos do governo através de redução de impostos sobre produtos industrializados do setor automobilístico [12].

Neste projeto nos restringimos ao aspecto da má qualidade dos serviços públicos no setor e propomos uma solução para tentar minimizar os sintomas causados pela ineficiência do transporte público.

## 2.2 Dados abertos

Diversas cidades do mundo vêm disponibilizando seus dados na Internet, as vezes para cumprir requisitos de transparência, outras para incentivar a participação popular e inovações, ou também para a criação de novos serviços ou novos conhecimentos criados a partir da combinação de dados de diversas fontes.

Dados abertos são dados disponibilizados livremente aos quais qualquer indivíduo possa ter acesso, possa modificá-los ou compartilhá-los para qualquer finalidade, estando sujeitos no máximo a exigências que visem preservar a sua proveniência e sua abertura [3].

Três leis foram propostas para os Dados Abertos Governamentais, mas é possível dizer que se aplicam a dados abertos em geral, são elas:

1. Se o Dado não pode ser encontrado e indexado na Web, ele não existe;
2. Se não estiver aberto e disponível em formato compreensível por máquina, ele não pode ser reaproveitado;
3. Se algum dispositivo legal não permitir sua replicação, ele não é útil.

Um grupo de 30 pessoas [4] se reuniu na Califórnia para definir os princípios dos Dados Abertos, princípios estes que juntamente com as leis, pautam os dados abertos. O grupo chegou a um consenso sobre oito princípios:

1. Completos. Todos os dados públicos são disponibilizados. Dados são informações eletronicamente gravadas, incluindo mas não se limitando a, documentos, bancos de dados, transcrições e gravações audiovisuais. Dados públicos são dados que não estão sujeitos a limitações válidas de privacidade, segurança ou controle de acesso, reguladas por estatutos;
2. Primários. Os dados são publicados na forma coletada na fonte, com a mais fina granularidade possível, e não de forma agregada ou transformada;
3. Atuais. Os dados são disponibilizados o quão rapidamente seja necessário para preservar o seu valor;
4. Acessíveis. Os dados são disponibilizados para o público mais amplo possível e para os propósitos mais variados possíveis;
5. Processáveis por máquina. Os dados são razoavelmente estruturados para possibilitar o seu processamento automatizado;
6. Acesso não discriminatório. Os dados estão disponíveis a todos, sem que seja necessária identificação ou registro;
7. Formatos não proprietários. Os dados estão disponíveis em um formato sobre o qual nenhum ente tenha controle exclusivo;

8. Livre de licenças. Os dados não estão sujeitos a regulações de direitos autorais, marcas, patentes ou segredo industrial. Restrições razoáveis de privacidade, segurança e controle de acesso podem ser permitidas na forma regulada por estatutos.

Utilizando-se desses dados abertos, é possível desenvolver sistemas para processar e exibir os dados de maneira mais amigável para os usuários, como é o caso dos aplicativos "Reclamações Procon" [6], ou o "Para onde foi o meu dinheiro" [5]. Ambos utilizam os dados disponibilizados pelas plataformas de dados abertos.

### **2.3.1 Dados abertos na Cidade do Rio de Janeiro**

O Portal de dados abertos da Prefeitura do Rio [7] foi criado para disponibilizar os dados abertos do município, funcionando como um grande catálogo, com o objetivo de facilitar a busca pelos dados publicados pela Prefeitura do Rio de Janeiro. Nele podemos encontrar dados de transporte e mobilidade, educação, turismo, meio ambiente e saúde, entre outros. Para o nosso projeto, utilizamos os dados de transporte e mobilidade, como os dados dos GPS instalados nos ônibus [8] que fazem parte da frota das concessionárias da Cidade do Rio de Janeiro, pontos de parada das linhas de ônibus [9] e pontos dos trajetos das linhas de ônibus [10].

Alguns dados, como os dos GPS instalados nos ônibus, podem ser atualizados com frequência e é possível verificar a frequência dessa atualização no próprio portal. Os dados de GPS são atualizados a cada minuto, enquanto os dados dos pontos de parada das linhas de ônibus foram atualizados pela última vez em 04 de Abril de 2014, no momento desta declaração. Estes dados são disponibilizados em diferentes formatos, dependendo do dado desejado, como JSON, CSV ou PDF.

### **2.3 Aplicativos de mobilidade urbana**

Para tentar compreender as necessidades dos usuários que procuram ferramentas para tentar melhorar a qualidade de seu transporte, faremos uma análise sobre alguns dos aplicativos mais usados pelas pessoas quando o assunto é mobilidade.

Serão três os aplicativos analisados, "Vá de ônibus" (cerca de 50 mil downloads, classificação 2.7 de 5), Moovit (cerca de 10 milhões de downloads, classificação 4,3) e Google Maps (cerca de 1 bilhão de downloads, classificação 4,3).

Existem outras aplicações com o mesmo propósito, mas após pesquisa na Internet concluímos que essas três opções eram as mais populares e completas quando olhamos para o município do Rio de Janeiro. Faremos um teste com os três aplicativos, que consistirá em procurar uma rota saindo da Avenida Rio Branco, no Centro do Rio, até a Avenida Pasteur, próximo ao 485.

### **2.3.1 Vá de Ônibus**

Desenvolvido pela própria Fetranspor (Federação das Empresas de Transportes de Passageiros do Estado do Rio de Janeiro), a proposta, segundo a descrição do aplicativo na Play Store, é calcular rotas, localizar pontos e identificar as linhas.

A primeira impressão que tivemos ao abrir o aplicativo foi positiva. É possível ver os ônibus que estão próximos à nossa localização. Podemos também visualizar todos os pontos de ônibus nas proximidades e quais linhas passam por cada ponto.

Para consultar uma rota, basta completar os dados de origem e destino. Neste sentido, o aplicativo é simples e de fácil compreensão. Porém, ao preencher uma origem ou destino, o aplicativo nos fornece opções fixas para a localização de chegada (ou partida), não sendo um campo de livre entrada, de modo que não podemos, por exemplo, marcar que desejamos parar próximo ao *485 da Avenida Pasteur*, ficando informado apenas *Avenida Pasteur*.

A apresentação dos resultados é relativamente fácil de ser compreendida. Porém, o resultado da pesquisa das rotas não foi muito satisfatório, talvez justamente pela limitação em relação ao preenchimento dos locais de origem e destino, não sendo possível informar o número do logradouro dentro da via de destino.

Entre as rotas diretas, foram dadas as opções das linhas 105, 106 e nas rotas com comutação entre duas linhas, 011 e 2014 ou 011 e 104. Sabemos que na Avenida Pasteur próximo ao 485, temos linhas que passam mais próximas do que as sugeridas, como a 107, 581, 582 e 513.

Outro problema encontrado no aplicativo foi, após executar esse processo e voltar para a tela inicial, que exibe os ônibus que estão nas proximidades, pudemos

notar que a posição dos ônibus não foi atualizada: continuava exibindo os mesmos ônibus, nos mesmos lugares de antes.

### **2.3.2 Moovit**

No primeiro acesso ao Moovit, a primeira impressão que tivemos foi de confusão. Logo na tela inicial somos apresentados a três opções: planejar uma viagem, estações ao redor ou encontrar uma linha. As opções de "planejar uma viagem" ou "encontrar uma linha" pareceram bastante similares. Por fim, descobrimos que para buscar uma rota, a opção desejada é a de "planejar uma viagem".

A partir daí a usabilidade é bastante intuitiva, respondendo a simples pergunta “Para onde você quer ir?”. Se o usuário permitir ao aplicativo acessar seus dados do GPS, ele já preenche as informações de origem, sempre permitindo a alteração manual. Depois, basta um clique para encontrar as possíveis rotas para alcançar o destino.

O resultado da pesquisa por rotas partindo da Avenida Rio Branco e indo para a Avenida Pasteur, próximo ao 485 retornou resultados satisfatórios, oferecendo comutações entre as linhas 474 e 107 ou 582, 457 e 107, ou ainda 721D e 513. Sabemos que as segundas linhas sugeridas são de fato respostas satisfatórias, pois as linhas 107, 582 e 513 de fato passam pelo ponto mais próximo ao prédio 485 na Avenida Pasteur.

### **2.3.3 Google Maps**

O Google Maps não é apenas um aplicativo para buscar rotas, principalmente rotas utilizando apenas o transporte público. Por isso, sua interface é menos simples de utilizar com o propósito de encontrar uma rota entre dois locais dentro da cidade do Rio de Janeiro. Mesmo assim, seu uso é bastante simples.

Para encontrar rotas de um ponto a outro na cidade é preciso fazer uma pesquisa no aplicativo. Em seguida, o aplicativo dará a opção de ir até o local pesquisado, calculando rotas de carro, transporte público, a pé ou de bicicleta, entre outras opções, dando ainda a possibilidade do usuário alterar a localização de partida. O Google Maps considera inicialmente que o usuário estará partindo no momento exato da pesquisa, estimando também o tempo que levará para o transporte passar. O horário de partida também é flexível e podemos programá-la para um momento mais oportuno.

Fazendo a consulta do caso de teste, partindo da Avenida Rio Branco a caminho da Avenida Pasteur, próximo ao 485, o resultado oferecido pelo o aplicativo não foi satisfatório. Uma das sugestões oferecidas é pegar o metrô até Botafogo e seguir a pé (uma caminhada de 24 minutos, pela estimativa do aplicativo) para a Urca. As outras opções sugeridas foram as linhas 104 e 105, que também incluem uma caminhada de mais de 20 minutos entre o ponto de descida até o destino desejado.

## **2.4 Considerações finais**

Neste capítulo apresentamos o conceito de mobilidade urbana e falamos do porquê a mobilidade urbana é um desafio nas grandes cidades. Explicamos também o que são dados abertos e quais são as leis e princípios que definem este tipo de dados, principalmente os governamentais. Também falamos do Portal de dados abertos da Prefeitura do Rio, portal de onde buscamos os dados utilizados neste projeto, além de uma breve análise dos aplicativos existentes na área de mobilidade urbana.

Também concluímos que, dos três aplicativos analisados, o mais preciso para o município do Rio de Janeiro é o Moovit, tanto o Vá de Ônibus quanto o Google Maps apresentaram soluções não ideais, e mais, os resultados de ambos foi bastante parecido, apesar da diferença entre as consultas.

No próximo capítulo apresentaremos a proposta de trabalho com os dados dos GPS dos ônibus, seus pontos de parada e trajetos e mostraremos o desenvolvimento da proposta até seu estágio final, bem como das dificuldades encontradas no caminho.

## 3 Solução Proposta

### 3.1 Um histórico da proposta

O objetivo desse projeto foi alterado devido a alguns problemas encontrados durante o desenvolvimento. Inicialmente, a proposta era desenvolver um sistema que, dados um ponto de origem e um ponto de destino, informaria ao usuário em quanto tempo um ônibus de uma linha que o deixasse perto do ponto de destino chegaria no ponto de parada mais próximo, de modo que o mesmo pudesse se programar para não perder o ônibus.

O cálculo para determinar o tempo de chegada esperado de um ônibus até um ponto de parada não é trivial. Após algumas considerações, chegamos a conclusão que decompondo o trajeto dos ônibus em pedaços pequenos e calculando uma série histórica de velocidade média em um determinado horário do dia seria possível prever com aproximação satisfatória o tempo de chegada do ônibus no ponto de parada.

O primeiro problema surge neste momento. Para implementar o algoritmo de previsão do tempo de chegada é essencial o conhecimento das rotas das linhas de ônibus. Ao analisar os dados disponíveis referentes às linhas dos ônibus verificamos que não possuímos documentos com pontos de parada de muitas linhas: cerca de 10% das linhas observadas não tinham documento de rota e cerca de 40% das linhas não possuíam documento com pontos de parada.

Como estes dados são vitais para a implementação do projeto, tivemos que repensar a nossa abordagem. Poderíamos usar o documento que é atualizado a todo minuto, contendo a posição georeferenciada de todos os ônibus que estão em circulação no momento, para tentar recriar as rotas das linhas. A vantagem dessa abordagem é que, independentemente dos documentos disponíveis, sempre seria possível atualizar a rota de uma linha a partir de um número suficiente de leituras. Considerando o cenário do transporte público no período em que este projeto foi realizado, em meio a obras na cidade, mudança nos itinerários, linhas que deixaram de existir e novas linhas que foram criadas, esta seria uma opção interessante. Entretanto, sua implementação não é simples.



Sabendo que os dispositivos GPS podem apresentar falhas, os motoristas dos ônibus podem optar por fazer trajetos alternativos em determinadas circunstâncias, as obras podem causar inconsistências nas leituras de um dia para o outro, o que dificultaria a decisão de como calcular as rotas de modo automatizado e com o mínimo de interferência humana possível, além da dificuldade que teríamos para checar a acurácia do algoritmo criado com todas essas alterações nos itinerários, seria difícil garantir que o algoritmo está gerando as rotas de maneira correta.

Vendo a complexidade que teria para primeiro criar um algoritmo para corrigir o problema das rotas não existentes e, em seguida, outro para calcular a velocidade média em um determinado trecho em um determinado horário, fazer uma série histórica dessas informações para serem analisadas futuramente e, finalmente, implementar a proposta, optamos por alterar o escopo do projeto. Ao invés de focar no tempo de chegada de determinado ônibus ao ponto de parada, o sistema passaria a informar quais combinações de linhas o usuário poderia utilizar para chegar ao seu destino, considerando sua posição atual [8].

Ainda precisávamos resolver o problema de encontrar os pontos de parada das linhas que não possuem documento com estas informações. Uma idéia seria identificar, nos pontos de parada conhecidos, quais as categorias de BRS do ponto, fazer a identificação da categoria do BRS em todas as linhas e usar o documento contendo a posição georeferenciada dos ônibus para deduzir quais linhas passam por determinado ponto. Essa dedução a princípio seria feita utilizando a proximidade das leituras do GPS dos ônibus em relação à posição georeferenciada do ponto. Um limite de distância máxima poderia ser utilizado para saber se um ônibus passa por um ponto.

Por fim, não foi preciso implementar essa solução, pois apesar de não ser disponibilizado no sistema de dados abertos do Rio de Janeiro um documento específico para todas as linhas de ônibus (muitas possuem, mas nem todas), encontramos um único documento com todos os pontos de parada para todas as linhas de ônibus.

### **3.1.2. Tecnologias usadas e armazenamento dos dados**

Para o desenvolvimento deste projeto, decidimos usar a linguagem Node.js, por motivos de praticidade e considerando o conhecimento prévio na mesma, o projeto poderia iniciar rapidamente, além de ser uma linguagem simples de ser implementada

em uma plataforma web para acesso remoto dos usuários. Optamos por um design simples, com uso superficial de HTML apenas para criar uma página que é usada para exibir o mapa, onde todas as informações são disponibilizadas e o mapa foi gerado através da API JS do Google Maps. Usamos também Javascript para programar a interação do usuário com a aplicação na parte do front-end.

No armazenamento de dados, inicialmente pensamos em usar o MongoDB devido a sua característica de armazenamento em JSON, que integra perfeitamente com o Javascript (usado tanto no front como no back-end).

Porém após constatarmos que a consulta a uma base externa não teria grandes vantagens em relação ao desempenho das buscas (até onde minha experiência com o MongoDB me permitiu descobrir, no momento do desenvolvimento do projeto, não encontrei informações sobre como indexar campos geoespaciais), o volume de dados ser pequeno e não justificar a persistência e a necessidade de processar esses dados de diversas maneiras diferentes ao longo do desenvolvimento, não havendo uma estrutura definitiva até próximo à conclusão do projeto, resolvemos manter os dados da aplicação em memória, processando os dados abertos fornecidos ao iniciar a aplicação, e daí em diante temos todos os dados processados a pronta disponibilidade na própria memória da aplicação.

É possível que uma pesquisa mais focada na questão dos índices geoespaciais no MongoDB, ou a utilização de um banco como o PostgreSQL, teria revertido em consultas mais simples no sistema e portanto a não necessidade de tanto processamento nos dados, pois a alta complexidade das buscas por proximidades no conjunto de pontos de parada dos ônibus tornar essa consulta muito ineficiente e lenta, foi preciso uma mudança na maneira de arranjar os dados.

### **3.2 A Implementação da proposta final**

Com a proposta do projeto alterada, agora voltada a indicar possibilidades de comutação para que o usuário chegue ao destino desejado e o trajeto a ser feito por ele, indicando onde poderia ser feita a comutação para a outra linha, e com o problema da falta dos dados dos pontos de parada de grande parte das linhas de ônibus resolvido,

faltava apenas implementar o algoritmo que resolveria as combinações possíveis de linhas para levar o usuário ao destino. Este algoritmo consiste de três etapas:

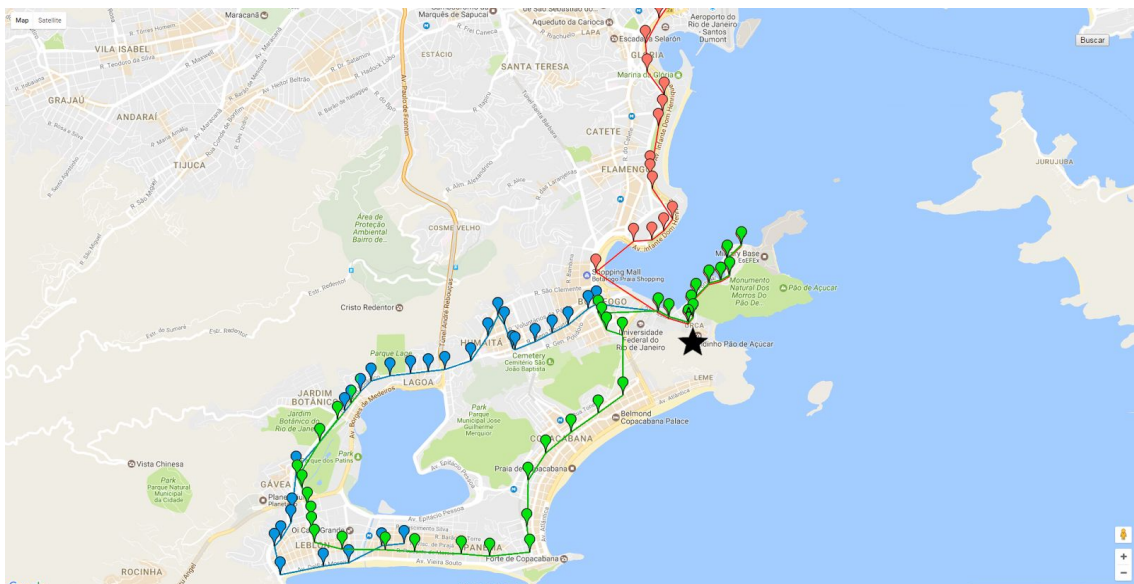
1. Buscar os pontos próximos ao destino desejado e identificar todas as linhas que têm pelo menos um daqueles pontos na sua lista de pontos de parada;
2. Buscar os pontos próximos ao local de saída e identificar todas as linhas que têm pelo menos um daqueles pontos na sua lista de pontos de parada;
3. Se encontrar uma linha na etapa anterior que tenha sido encontrada na primeira etapa, esta já é uma solução (trajeto sem comutação);
4. Se não encontrar, iterar em todas as linhas que passam pelos pontos próximos ao local de saída e, para cada linha, iterar em todos seus pontos seguintes, verificando em cada ponto analisado se uma das linhas que passam por ele também passa no ponto próximo ao destino desejado.

Uma das premissas que consideramos na concepção desse algoritmo foi que limitaríamos o número máximo de linhas que entrariam na composição: seriam duas, ou seja, seria permitido no máximo uma comutação. Esta restrição foi adotada por dois motivos. Primeiro, porque existe essa limitação no bilhete único, que permite a integração entre, no máximo, dois ônibus. Segundo, pelo aumento da complexidade do código exigido para verificar as possibilidades com maior número de integrações, que se feita de modo simplista poderia ocasionar um maior tempo de resposta ao usuário ou sobrecarga no sistema, que poderia acarretar em problemas de disponibilidade do serviço. Portanto, nos parece razoável assumir o limite máximo de uma integração entre duas linhas de ônibus.

Para implementar o algoritmo, fizemos uma alteração na estrutura dos dados fornecidos pelo Dados Rio. Ao invés de guardar os dados como são fornecidos no documento (organizados por linha e, dentro de cada linha, por ponto de parada), fizemos uma decomposição desses dados em pontos de parada e linhas. A estrutura que representa cada ponto de parada contém sua localização georeferenciada (latitude e longitude), uma lista das linhas que passam por este ponto e os outros pontos de parada

próximos a ele. Essa estrutura é utilizada para agilizar o processo de busca das comutações possíveis para chegar ao destino desejado.

Considerando os dados são fornecidos, para encontrar opções de comutação em cada ponto de parada de cada linha partindo do ponto da origem, seria necessário percorrer a lista de pontos de cada linha de ônibus, verificar a proximidade do ponto analisado e, sendo próximo o suficiente, marcar a linha daquele ponto de parada como uma opção de comutação. Como nenhuma ordenação contribuiria para reduzir o tempo de busca com esta estrutura de dados, executar o algoritmo para cada pesquisa seria custoso para o servidor e poderia levar muito tempo, inviabilizando essa opção.



*Figura 1 - Representação dos pontos das linhas que passam próximas ao ponto de origem (estrela preta)*

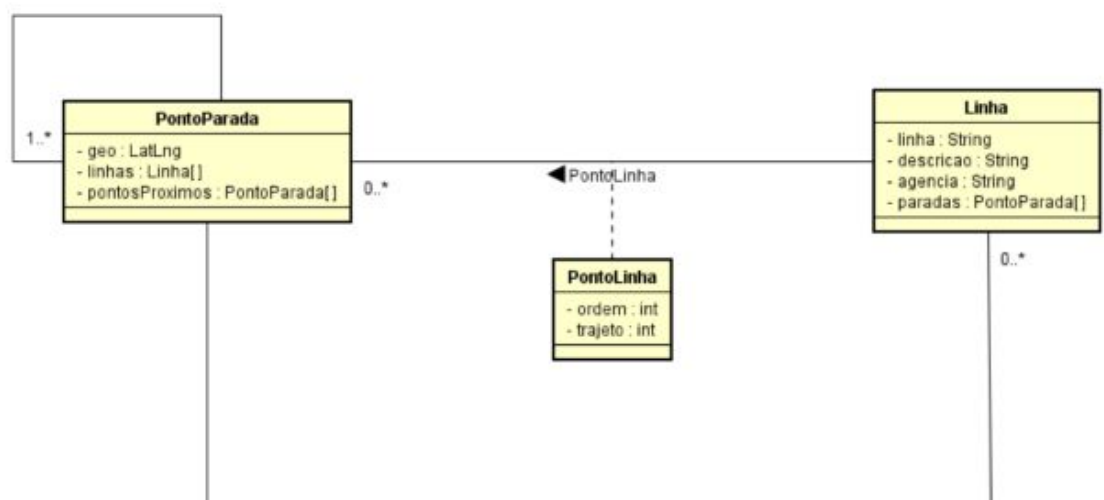
A Figura 1 ilustra o cenário apresentado nos parágrafos anteriores. A origem, que é representada na estrela preta, tem um único ponto próximo, por onde passam três linhas de ônibus. Cada balão representa um ponto de parada das três linhas e cores diferentes representam linhas distintas. Utilizando os dados com a mesma estrutura com que são fornecidos, para encontrar as comutações para essas três linhas precisaríamos percorrer todo o arquivo de pontos de parada para cada um dos pontos, procurar os pontos próximos e verificar se as linhas que passam próximas ao destino estão nos pontos ou próximo a eles. Em números, considerando que cada linha tenha 50 pontos de parada, precisaríamos verificar 150 pontos. O documento de pontos de parada tem

62.033 registros e precisaríamos percorrer essa lista 150 vezes a procura dos pontos de parada próximos ao ponto analisado para cada pesquisa do usuário.

Chegamos à conclusão que o ideal seria armazenar os dados de cada ponto em uma estrutura que facilitasse a consulta, guardando uma lista de todas as linhas de ônibus que passam por este ponto e os pontos de parada que ficam a um certo raio de distância dele. Fariamos então a detecção dos pontos próximos apenas uma vez, reestruturariamos os dados armazenando todos os pontos de parada nesse novo formato, o que torna a consulta para buscar as possíveis linhas para completar o trajeto necessário quase imediata, bastando iterar nos pontos de parada de qualquer uma das linhas (passando no ponto de origem ou chegando no destino) e verificar se no ponto, ou nos pontos próximos a ele, passa uma das linhas que são possíveis respostas.

A estrutura das linhas armazena uma descrição da linha (como por exemplo, "Urca x Leblon"), a agência responsável pela linha e uma lista com os pontos de parada da linha. No caso das linhas não circulares, a estrutura armazena os pontos de parada tanto no trajeto de ida quanto no trajeto de volta. Já para as linhas circulares é mais simples, pois temos apenas um trajeto.

Para cada ponto de parada da linha temos uma informação adicional que é um número de sequência. Esse número representa a ordem dos pontos ao longo do trajeto percorrido pelos ônibus daquela linha, sendo um ponto com número de sequência '1', o primeiro ponto da linha, o ponto de partida. O ponto com número de sequência '2' é o próximo ponto pelo qual o carro passará, e assim sucessivamente.



Com os dados estruturados e a complexidade das buscas reduzida, ainda que com um pequeno custo adicional devido ao processamento necessário para preparar os pontos de parada na nova estrutura de dados, restava disponibilizar a consulta para os usuários através de uma interface simples. A implementação da interface seria composta por duas partes: primeiro, uma API RESTful para acessar os dados do sistema e fazer a consulta das possíveis comutações até o destino desejado, dado um ponto de partida; segundo, uma página web composta por um mapa, dois marcadores, um para o ponto de partida e o outro para o destino, e um botão que iniciaria uma ação para consumir a API criada e retornar as informações da consulta para o usuário, através do desenho da rota no próprio mapa, informando o ponto de comutação.

Após concluir o desenvolvimento da interface e testar o resultado dos algoritmos implementados, tivemos uma visão mais ampla da situação e percebemos uma falha no algoritmo que comprometia o seu resultado final. Esta falha, assim como sua correção, será discutida na próxima seção.

### **3.3 O problema dos dados disponibilizados**

Como explicado anteriormente, armazenamos todos os pontos de parada de uma linha em uma lista. Em cada ponto de parada temos um número de sequência (1, 2, 3, ..., n), representando a ordem dos pontos no trajeto da linha. Para cada linha não circular, temos um trajeto de ida e um de volta e, conseqüentemente, pontos com números de sequência repetidos (dois pontos '1', dois pontos '2', e assim por diante). Eventualmente, um trajeto de linha não circular, na ida ou na volta, poderia passar por mais pontos de parada do que o outro trajeto, de modo que ficamos com várias duplicidades nos números de sequência. Quando o número de pontos é igual nos dois trajetos, temos duplicatas de número de sequência para todos os pontos.

O problema que comprometeu o resultado do sistema foi não haver informação que diferenciase um trajeto de outro. Com dois pontos '1', como saberíamos encontrar o ponto '2' seguinte ao ponto desejado? Até então, não tínhamos levado esse problema em consideração. Por exemplo, um cenário complicado seria pegar um ônibus no centro da cidade, no Instituto Biomédico da UNIRIO, desejando ir até o IMPA (Instituto Nacional de Matemática Pura e Aplicada) no Jardim Botânico. Neste caso, precisaríamos pegar

dois ônibus, uma opção sendo uma comutação entre as linhas 107 e 409. Digamos que o ponto de parada mais próximo ao ponto de partida seja um ponto de parada da linha 107 com número de sequência 5. Então, deveríamos percorrer os pontos com número de sequência superior a 5 (6, 7, 8, ...) e verificar se alguma linha que passe perto do destino desejado também passa por esses pontos, como a linha 409.

Entretanto, por não sabermos se estamos no trajeto de ida ou de volta da linha em questão olhando apenas para o número de sequência, podemos cometer o erro de olhar para um ponto que não é do mesmo trajeto, e sim do trajeto de sentido oposto, sendo necessário assim ir até o ponto final e continuar na mesma linha após essa começar seu novo trajeto. Supondo que o ponto de partida faça parte do trajeto de ida da linha 107, precisamos saber qual ponto com número de sequência pertence ao mesmo trajeto de ida e olhar apenas para os pontos com numeração superior que pertencem também a este trajeto, desconsiderando os pontos do trajeto de volta. Ou seja, precisamos adicionar uma nova informação na estrutura que armazena os pontos de parada: o trajeto a que o ponto pertence, podendo ser de ida ou volta.

Para resolver os problemas das linhas circulares, a abordagem é simples: (i) verificamos se existem duplicidades dos números de sequência; (ii) se não houver duplicidades, a linha é circular e basta indicar que todos os pontos pertencem ao mesmo trajeto (ida, por exemplo). No caso das linhas não circulares, chegamos a uma solução que parecia simples e eficaz: (i) montar uma lista com todos os pontos de parada da linha; (ii) buscar os dois pontos que possuem número de sequência igual a 1; (iii) selecionar um destes pontos (ponto de referência) e atribuir ao trajeto de ida; (iv) verificar qual ponto com número de sequência seguinte (no caso, número de sequência '2') se localiza mais próximo ao ponto de referência; (v) atribuir o ponto mais próximo do ponto de referência ao trajeto de ida; (vi) atribuir o ponto mais distante do ponto de referência ao trajeto de volta; e (vii) usar o ponto mais próximo como um novo ponto de referência e voltar ao passo (iv).

Após a implementação da solução, fizemos alguns testes usando as linhas 409 e 512, que serviam como controle para ajuste do algoritmo por serem linhas de rotas conhecidas. A verificação era feita plotando as rotas sobre um mapa, usando cores diferentes para os diferentes trajetos, e o resultado parecia bastante satisfatório.

Entretanto, ao olhar para outros casos foi possível verificar as limitações do algoritmo. Alguns trajetos não iam até o final -- ao invés disso, iam até metade do caminho e voltavam para o ponto de partida, de modo que parecia que determinadas linhas tenham dois trajetos distintos, isolados um do outro. Em outros casos, os erros eram mais difíceis de descrever, mas grande parte desses se devem a inconsistências no documento de pontos de parada. Por exemplo, a linha 6, que é descrita como uma linha circular, possui 3 pontos com o mesmo número de sequência entre 1 e 30. Do 31 até o 38 temos duplicidades nos pontos e a partir daí até o número de sequência 44 existe apenas um ponto por número. Outras linhas possuíam números de sequência únicos, não havendo sequer uma duplicata. Porém, ao desenhar esses pontos em ordem no mapa, pudemos notar que os pontos de parada subsequentes estavam com coordenadas incorretas, como um ponto próximo ao centro da cidade, o seguinte próximo à Barra da Tijuca e o seguinte a esse no Flamengo.

Após uma análise dos casos em que não havia erro no arquivo, observamos que quando os pontos médios dos trajetos se aproximavam, o ponto mais próximo ao ponto de referência poderia levar o trajeto de volta a seu ponto de partida. Em um exemplo que observamos, um ônibus seguia até o Terminal da Alvorada, na Barra da Tijuca, onde os pontos de número 33 dos trajetos de ida e volta se encontravam. Casualmente o ponto de número 34 do trajeto de volta estava mais próximo do ponto 33 do trajeto de ida e vice-versa, fazendo com que o ônibus voltasse exatamente ao ponto de onde partiu. Sendo assim, a linha parecia ter dois trajetos circulares independentes, que se encontravam no Terminal da Alvorada.

Com isso havia o risco de escolher o próximo ponto de parada incorretamente, sendo o ponto mais próximo ao analisado na realidade do outro trajeto, como podemos ver na Figura 2 que representa os trajetos calculados para a linha 107. A partir daí todo o trajeto fica comprometido, pois o ponto de número de sequência seguinte mais próximo continuará sendo o do trajeto errado e o problema não se corrigirá, gerando assim dois trajetos diferentes que se aproximam no meio do caminho, mas se afastam novamente ao definir erradamente os trajetos dos pontos.





*Figura 2 - Falha na identificação dos trajetos da linha 107*

Uma das soluções levantadas seria a possibilidade de, ao invés de iniciar a busca pelos trajetos do ponto inicial, partir tanto do ponto inicial quanto do ponto final de um trajeto. A determinação de qual dos pontos finais seria o ponto final do trajeto é relativamente segura, considerando que dos dois pontos possíveis como finais no mesmo trajeto, o ponto inicial e final devem ser os mais distantes, o que faz sentido se considerarmos que normalmente o ponto final de um trajeto é o ponto inicial no trajeto sentido contrário. Partindo então do ponto inicial e do ponto final ao mesmo tempo, procurando sempre os pontos mais próximos, quando chegássemos no ponto médio, bastaria unir as duas listas e o problema estaria resolvido.

Outra possível solução seria escolher um trajeto e definir seu ponto inicial e final, escolhendo os mais distantes, da mesma maneira que o procedimento anterior. Porém, ao invés de fazer uma busca a partir do início e outra do final, montar a rota escolhendo um ponto a ser analisado e procurar os pontos com número de ordem seguinte ao ponto analisado, marcar o mais próximo como ponto da mesma rota e repetir a análise com este ponto. Se encontrarmos dois pontos com número de ordem seguinte dentro de um certo raio do ponto analisado, marcamos o ponto analisado como um ponto de incerteza. Se não for possível completar a rota, isto é, chegar ao ponto final definido para o trajeto, voltamos nos pontos de incerteza e invertemos a escolha feita, até que consigamos completar o trajeto. Com este algoritmo, os dados de algumas linhas foram corrigidos, como a linha 107 que pode ser vista na Figura 3.

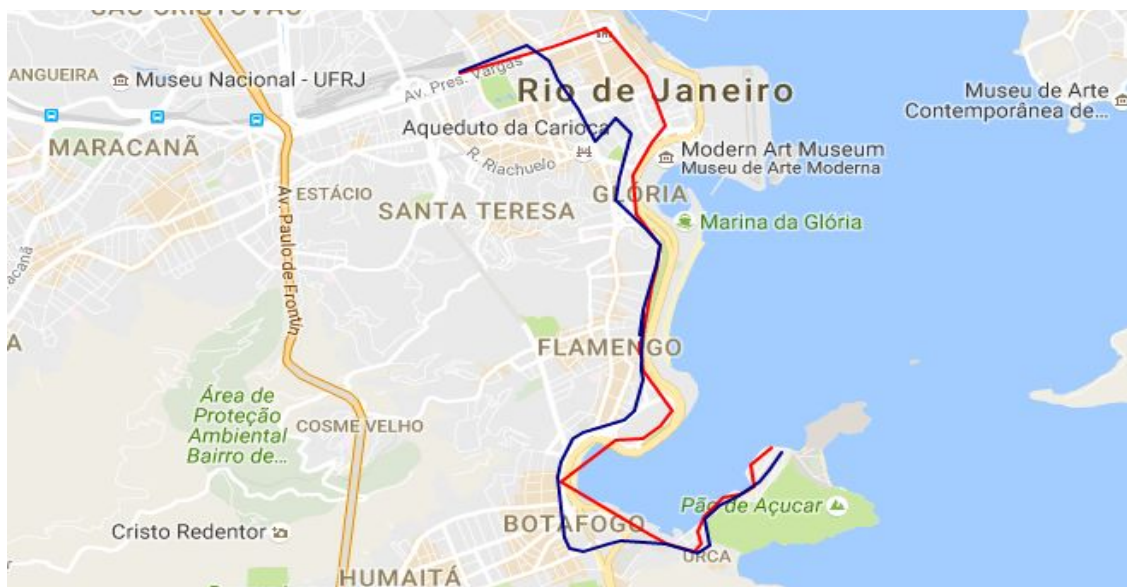


Figura 3 - Trajetos da linha 107 com os algoritmos de definição de trajetos.

Foram necessários alguns ajustes em relação a distância que deveria ser considerada para marcar um ponto como ponto de incerteza, por conta de termos linhas que passam por túneis e, dependendo da extensão do túnel, é possível que os pontos de parada estejam longe um do outro. Um exemplo do problema com os túneis foi encontrado após o processamento dos dados da linha 309. Como podemos ver na Figura 4, o trajeto se perde próximo ao Túnel Zuzu Angel.



*Figura 4 - Problema ao procurar o ponto mais próximo na linha 309 quando passa por um extenso trecho sem pontos.*

Também alteramos a abordagem em relação aos pontos de incerteza. Ao invés de apenas marcar os pontos, criamos uma pilha: sempre que encontramos um ponto de incerteza, escolhemos uma das opções e jogamos a outra na pilha. Se chegarmos ao mesmo número de sequência de um ponto final mas os pontos fossem distintos, removemos o último ponto adicionado da pilha e voltamos o trajeto até ele, repetindo o procedimento até atingirmos o ponto final correto ou a pilha ficar vazia. Optamos por ignorar as linhas quando a pilha ficar vazia. Após esses ajustes, tivemos um resultado bastante satisfatório, inclusive nos casos das linhas que passam por locais onde não existem pontos de parada próximos uns dos outros.

### **3.4 Considerações finais**

Com o problema dos trajetos resolvido, foi possível voltar a focar na questão central do projeto, que é a pesquisa das possíveis rotas de um ponto a outro. Exemplos de aplicação dos algoritmos aqui discutidos serão apresentados no próximo capítulo.

## 4 Exemplo de Uso

A fim de verificar a precisão dos algoritmos implementados, simularemos algumas pesquisas de rotas utilizando o nosso sistema e compararemos com outro aplicativo de mobilidade urbana. Como fizemos análises dos aplicativos de mobilidade urbana disponíveis no município do Rio de Janeiro, pudemos excluir os aplicativos que apresentaram resultados não ideais, de modo que faremos uma comparação dos resultados gerados pelo nosso sistema com os resultados providos pelo Moovit.

### 4.1 Projeto exemplo

Não está no escopo deste projeto o desenvolvimento de uma interface sofisticada com o usuário, mas a implementação e avaliação inicial dos algoritmos relacionados com a descoberta de rotas e comutações. Precisamos fazer uma interface para teste e uso para verificarmos a eficácia dos códigos implementados. Por isso, a interface do sistema (apresentada Figura 5) é composta por uma única tela, onde temos um mapa e dois marcadores que podem ser arrastados: um vermelho, representando o ponto de partida, e um verde, representando o destino desejado. Além disso, no canto superior direito temos um botão com o título "Buscar" para fazer a busca sobre as possíveis rotas entre os pontos selecionados.

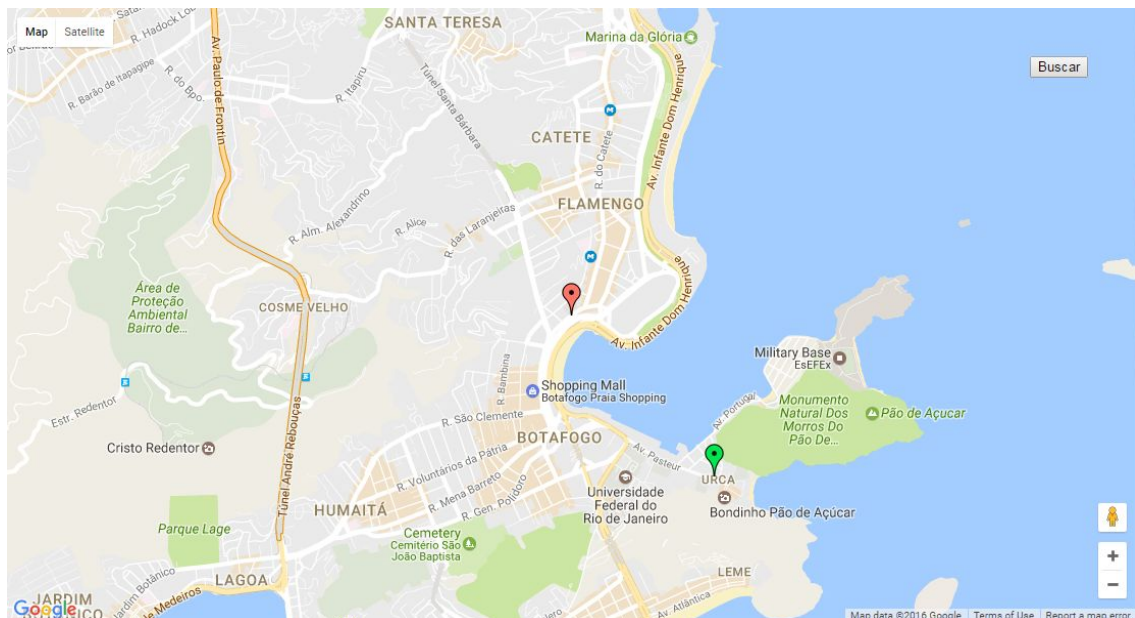


Figura 5 - Interface do sistema para consulta de rotas

Ao clicar no botão "Buscar", uma janela informa quais as linhas de ônibus que podem ser utilizadas para alcançar o destino (Figura 6). Temos dois tipos de rota: rotas diretas, onde não é preciso fazer comutação de ônibus, ou rotas com duas linhas, onde será preciso, em algum momento, trocar de ônibus.

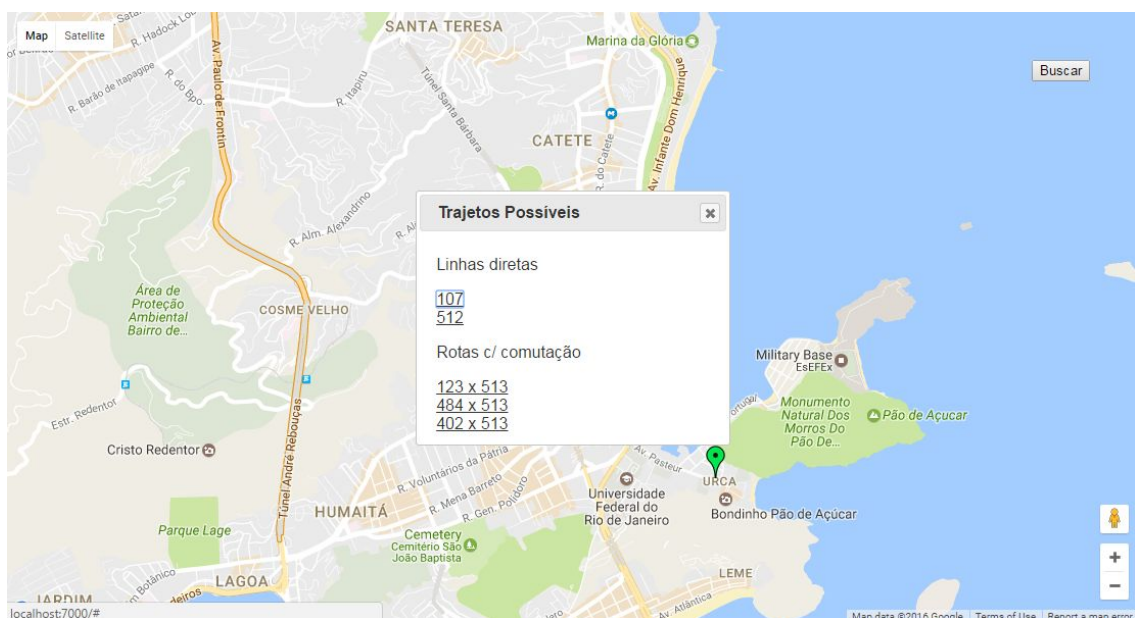
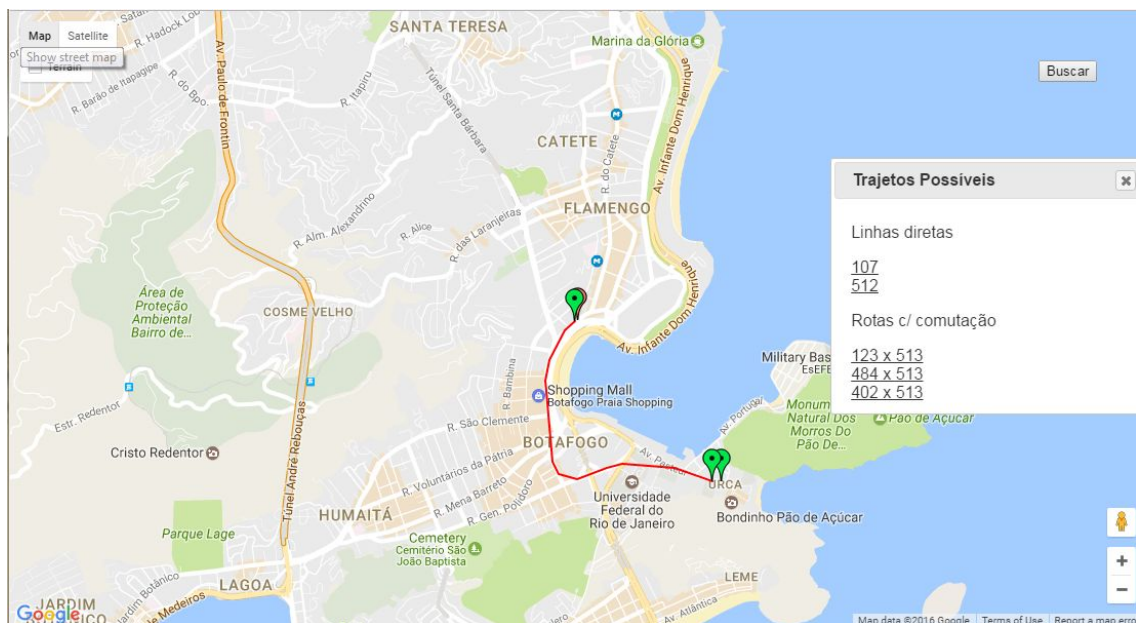


Figura 6 - Feita busca saindo da Praia de Botafogo direção Urca

A janela de resultados pode ser arrastada para permitir melhor visualização do mapa enquanto analisamos as rotas possíveis. Clicando nos números das linhas de



ônibus, a interface marcará os pontos de ônibus próximos ao ponto de origem e ao destino, para que o usuário possa ver a distância que deverá ser percorrida a pé.

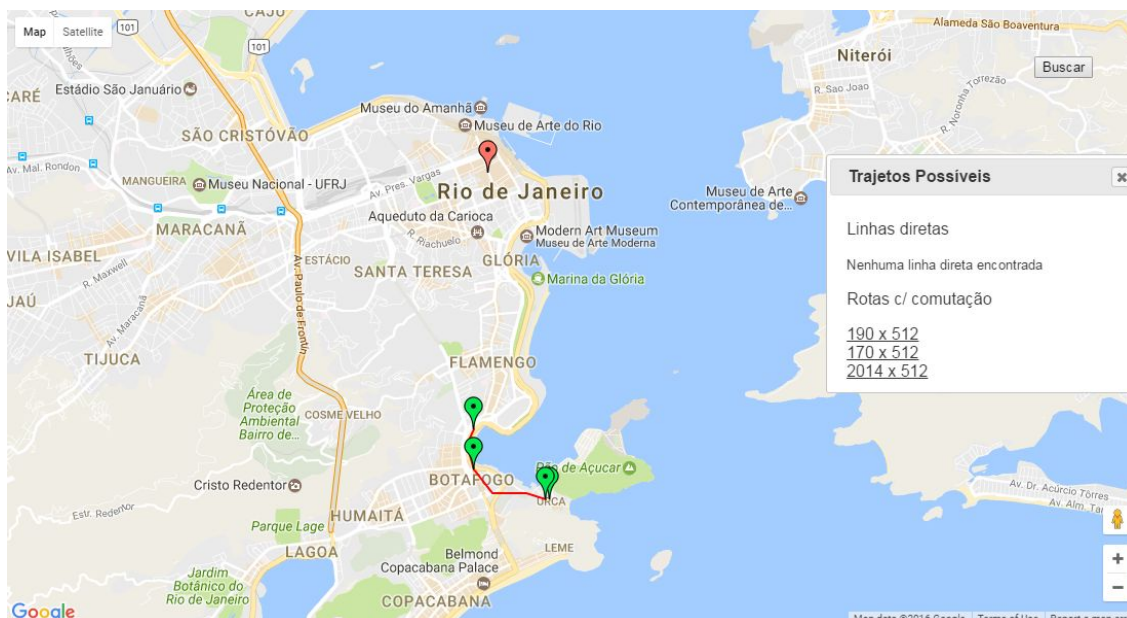


*Figura 7 - Selecionada a linha 107, partindo da Praia de Botafogo, próximo ao Edifício Argentina, em direção à Avenida Pasteur, próximo ao 485.*

Como podemos ver na Figura 7, além de marcar os pontos de ônibus, a interface apresenta uma linha com o trajeto aproximado que será percorrido pelo ônibus. O trajeto é aproximado, pois esta linha é formada pelos pontos de parada no meio do caminho, e pelo trajeto em si. Sendo assim, as curvas podem não se precisamente definidas por ele.

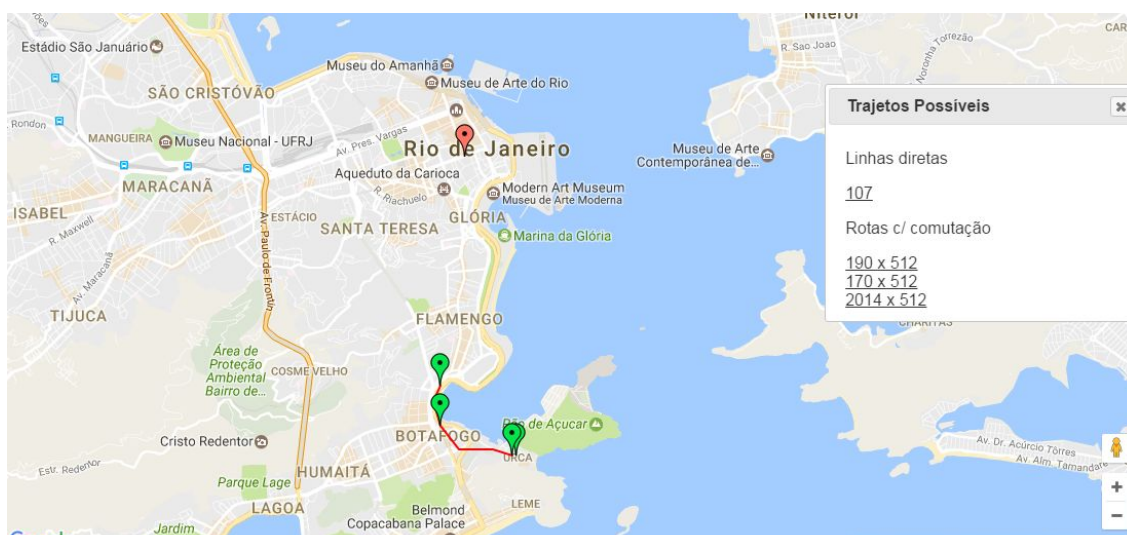
## 4.2 Exemplo de Uso do Sistema de Busca de Comutações

Para termos uma melhor percepção dos resultados gerados e das opções de rotas sugeridas pelo sistema, aumentamos um pouco a distância entre os pontos de origem e destino, tentando aproximar a análise do algoritmo proposto com os testes que foram realizados com os outros aplicativos.



*Figura 8 - Movido o marcador de seleção do ponto de origem. O mapa não é limpo depois de fazer uma nova pesquisa, mantendo a exibição da última rota selecionada*

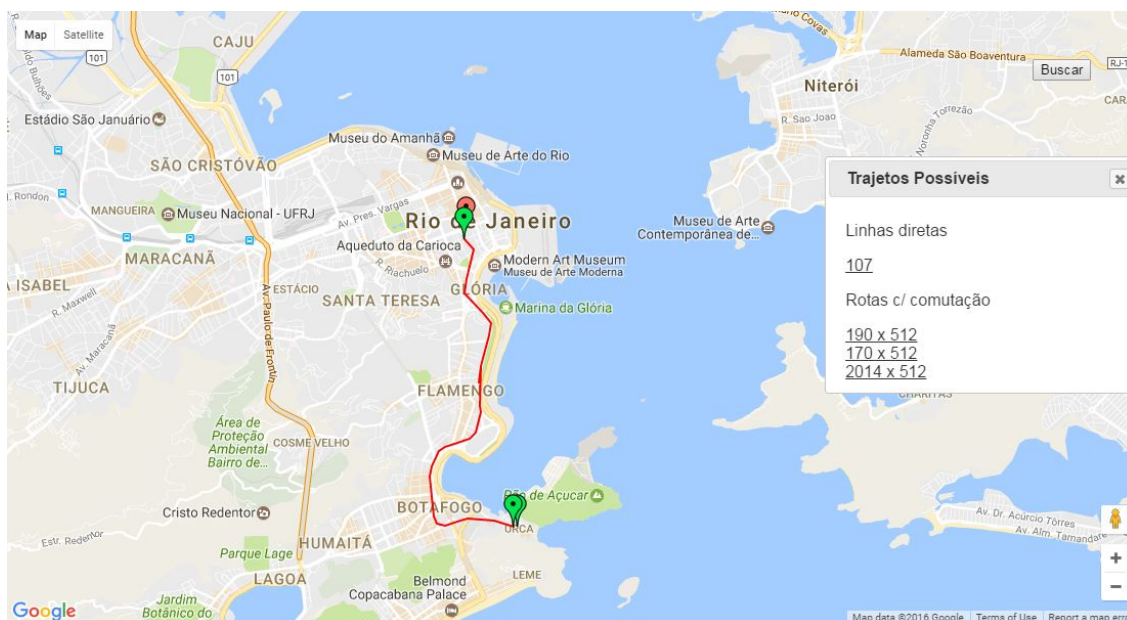
Podemos notar que, dependendo do local na Avenida Rio Branco de onde procuramos um ônibus para a Urca, as opções mudam por conta da distância do ponto de partida com o ponto de ônibus por onde passam as linhas que poderiam nos levar até o destino desejado (Figuras 8 e 9).



*Figura 9 - Ao mover o marcador mais para baixo, a opção da linha 107 aparece*

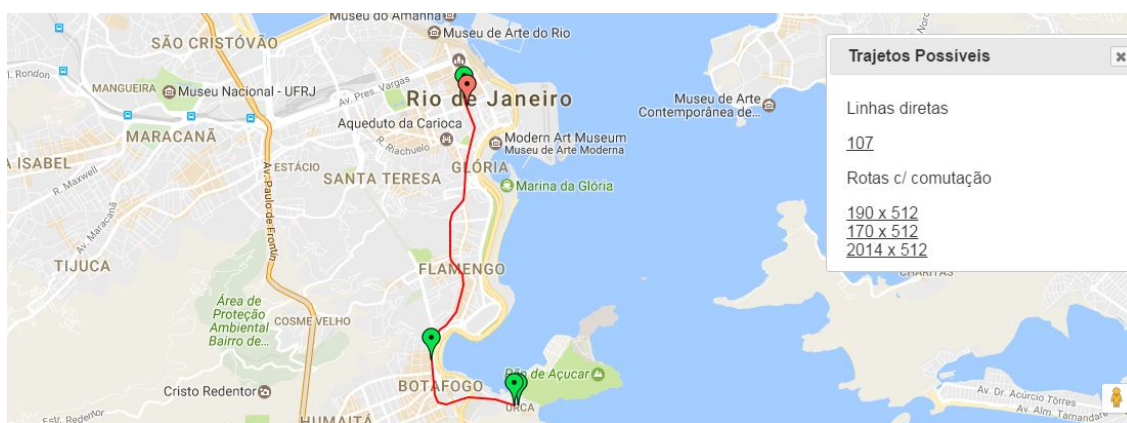
Vemos que as soluções apresentadas para ir da Avenida Rio Branco para a Avenida Pasteur, próximo ao 485, são bastante parecidas com os resultados oferecidos

pelo Moovit, e são também bastante satisfatórias. Clicando em uma das opções de rotas sugeridas, podemos ver o trajeto que será feito até o destino (Figura 10).



*Figura 10 - Selecionando o 107 para ir para a Urca do Centro da cidade*

A Figura 11 apresenta a opção por uma rota com comutação, no caso entre as linhas 170 (Rodoviária x Gávea) x 512 (Urca x Leblon). O Moovit apresenta uma opção parecida com essa, mas com os dados atualizados (após a extinção da linha 512, agora temos a linha CIR2 / 582), sugerindo pegar o 415 no mesmo local, na Avenida Rio Branco, e fazendo a baldeação em frente ao Shopping Botafogo, de modo que o trajeto é quase idêntico ao apresentado por nossos algoritmos.



*Figura 11 - Utilizando as linhas 170 e 512, fazendo a baldeação próximo ao Botafogo Praia Shopping.*



### **4.3 Considerações finais**

Como pudemos verificar, o sistema apresentou resultados semelhantes ao aplicativo Moovit, no caso de teste escolhido para avaliar os três aplicativos (Vá de Ônibus, Moovit e Google Maps). Ao iniciar os testes, imaginávamos que os melhores resultados seriam apresentados pelo Google Maps, devido ao tamanho da equipe envolvida e dos recursos mobilizados.

Ficamos surpresos ao constatar que não era o caso e que utilizando os dados abertos da Cidade do Rio de Janeiro conseguimos obter bons resultados (melhores que os gerados pelo Google Maps e compatíveis com os gerados pelo Moovit).

Se tivéssemos acesso a dados de maior qualidade, seria possível aumentar a precisão do sistema e garantir maior confiabilidade, pois ainda temos alguns problemas em relação a defasagem da informação, no que tange as novas linhas criadas recentemente, além de erros que não pudemos resolver com algumas linhas.

## 5 Conclusão

Os resultados do sistema de consulta a opções de rotas entre dois pontos dentro do município do Rio de Janeiro foram bastante satisfatórios. Em grande parte dos testes os resultados apresentados estavam de acordo com as opções informadas pelo Moovit. Pelo conhecimento de algumas linhas de ônibus da cidade, pudemos confirmar que seria possível usar as linhas sugeridas para chegar ao destino desejado.

Apesar disso, o sistema não está a prova de erros e muito menos completo. Devido à defasagem dos dados ofertados pelo Portal de Dados Abertos da Prefeitura da Cidade do Rio de Janeiro e a sua falta de integridade, muitas informações ainda estão faltando (rotas e pontos de parada), estão inconsistentes (linhas de ônibus antigas) e às vezes com erros (ônibus trafegando no mar ou na Baía de Guanabara).

Muito trabalho precisou ser feito para tentar tornar os dados úteis para a nossa proposta e corrigir todos os problemas deles seria uma tarefa muito complicada e custosa, envolvendo possivelmente trabalho de campo para fazer com que os dados representassem fielmente a realidade.

### 5.1 Contribuições

As principais contribuições deste projeto são: (i) apresentamos uma solução para o problema da falta de informação sobre sentido dos pontos de parada nos trajetos dos ônibus, ausente nos dados abertos fornecidos pela Prefeitura; e (ii) criamos um sistema que permite ao usuário descobrir qual ou quais linhas de ônibus ele deve pegar para ir de um ponto a outro dentro do município do Rio de Janeiro, informando o local da comutação, se houver.

### 5.2 Trabalhos futuros

Com o problema da identificação dos pontos de parada resolvido, é possível seguir adiante com a implementação de outras idéias, como a previsão do tempo até determinado ônibus passar por determinado ponto de parada. A criação de um algoritmo

para previsão do tempo aproximado para a próxima chegada de um ônibus em um ponto de parada abre portas para alguns projetos interessantes, como um sistema que informe a previsão de chegada dos ônibus nos pontos de parada. Essa idéia seria especialmente atraente considerando por exemplo, a utilização deste sistema em conjunto com um sistema de aviso sonoro para auxiliar deficientes visuais durante sua mobilidade.

Todo o código desenvolvido durante este projeto também está disponível em <https://github.com/urielb/riobuzz>, para ser usado como referência ou caso alguém queira contribuir com o projeto.

### **5.3 Limitações do estudo**

A principal limitação deste estudo foi o fato dos dados abertos disponibilizados pelo portal da Prefeitura do Rio de Janeiro não estarem completos, nem atualizados. Os dados dos pontos de parada das linhas de ônibus estão muito defasados, datando do início de 2014 e só nesse último ano tivemos diversas mudanças nos itinerários dos ônibus municipais. Além disso, faltam informações importantes sobre os pontos de parada das linhas, como a qual trajeto o ponto de parada pertence, de modo que não era possível, originalmente, alinhar corretamente os pontos em sequência.

Por conta dos problemas encontrados, tivemos que rever o escopo e alterar a proposta inicial, pois com o esforço necessário para resolver os problemas encontrados em relação aos dados não teríamos tempo de implementar a proposta inicial.

## Referências

- [1] IPEA (2011). *SIPS – Sistema de Indicadores de Percepção Social – Mobilidade Urbana*. Brasília, IPEA. Disponível em: [http://www.ipea.gov.br/portal/images/stories/PDFs/livros/livros/livro\\_sistemaindicaadores\\_sips\\_01.pdf](http://www.ipea.gov.br/portal/images/stories/PDFs/livros/livros/livro_sistemaindicaadores_sips_01.pdf). Acesso em 4 dez. 2016
- [2] GOLLNICK, Sergio - Viver Urbanamente. Disponível em: <http://gollnick.blog.terra.com.br/category/dicionario-do-urbanismo-de-joinville/>. Acesso em: 2014
- [3] PORTAL BRASILEIRO DE DADOS ABERTOS - O que são dados abertos? Disponível em <http://dados.gov.br/dados-abertos/>. Acesso em 4 dez. 2016
- [4] The 8 Principles of Open Government Data. Disponível em: <https://opengovdata.org/>. Acesso em 8 de dez. 2016
- [5] GASTOS ABERTOS - Para onde foi o meu dinheiro? Disponível em <http://paraondefoi.omeudinheiro.org.br/sobre>. Acesso em 4 dez. 2016
- [6] RECLAMAÇÕES PROCON. Disponível em <http://www.reclamacoesprocon.com.br>. Acesso em 4 dez. 2016
- [7] DATA.RIO - Portal de dados abertos da Prefeitura do Rio. Disponível em <http://data.rio/>. Acesso em 4 dez. 2016
- [8] DATA.RIO - Posições GPS dos ônibus em CSV. Disponível em <http://dadosabertos.rio.rj.gov.br/apiTransporte/apresentacao/csv/onibus.cfm>. Acesso 4 dez. 2016
- [9] DATA.RIO - Pontos de parada de todas as linhas de onibus. Disponível em [http://dadosabertos.rio.rj.gov.br/apiTransporte/Apresentacao/csv/gtfs/onibus/paradas/gtfs\\_todas-linhas-paradas.csv](http://dadosabertos.rio.rj.gov.br/apiTransporte/Apresentacao/csv/gtfs/onibus/paradas/gtfs_todas-linhas-paradas.csv). Acesso 4 dez. 2016
- [10] DATA.RIO - Pontos dos trajetos das linhas de ônibus. Disponível em <http://data.rio/dataset/pontos-dos-percursos-de-onibus>. Acesso 4 dez. 2016
- [11] COSTA; DA SILVA; COHEN. A origem do caos: A crise de mobilidade no Rio de Janeiro e a ameaça à saúde urbana, dez. 2013. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S2236-99962013000200411](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2236-99962013000200411). Acesso 4 dez. 2016

- [12] PENA, Rodolfo F. Alves. "Mobilidade urbana no Brasil"; *Brasil Escola*. Disponível em <http://brasilescola.uol.com.br/geografia/mobilidade-urbana-no-brasil.htm>. Acesso em 03 de dezembro de 2016.