Pontifícia Universidade Católica
do Rio de Janeiro

Rafael Barbosa Nasser

Uma Plataforma na Nuvem para Armazenamento de Dados Georreferenciados de Mobilidade Urbana

Tese de Doutorado

Tese apresentada ao Programa de Pós-graduação em Informática da PUC-Rio como requisito para obtenção do título de Doutor em Informática.

Orientador: Prof. Hélio Côrtes Vieira Lopes

Rio de Janeiro
Setembro 2016

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Rafael Barbosa Nasser

Uma Plataforma na Nuvem para Armazenamento de Dados Georreferenciados de Mobilidade Urbana

Tese apresentada como requisito parcial para a obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Hélio Côrtes Vieira Lopes
Orientador
Departamento de Informática – PUC-Rio

Prof. Carlos José Pereira de Lucena
Departamento de Informática – PUC-Rio

Prof. Marco Antonio Casanova
Departamento de Informática – PUC-Rio

Prof. Marcos de Oliveira Lage Ferreira
UFF

Prof. José Antonio Fernandes de Macêdo
UFC

Prof. Márcio da Silveira Carvalho
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 26 de setembro de 2016

Rafael Barbosa Nasser

O autor é Mestre em Informática e graduado em Engenharia de Computação pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) com domínio adicional em Empreendedorismo pela mesma Universidade. Atua como Coordenador Técnico do Laboratório de Engenharia de Software da PUC-Rio, tendo mais de 10 anos de experiências em projetos na área de tecnologia da informação.

Ficha Catalográfica

Nasser, Rafael Barbosa

Uma Plataforma na Nuvem para Armazenamento de Dados Georreferenciados de Mobilidade Urbana / Rafael Barbosa Nasser ; orientador: Hélio Côrtes Vieira Lopes – 2016.

159 f. : il. (color.) ; 30 cm

Proposta de Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2016.

Inclui bibliografia

1. Informática – Teses. 2. Data Science. 3. Computação na nuvem. 4. Computação paralela. 5. Geoprocessamento. 6. Engenharia de software. I. Lopes, Hélio Côrtes Vieira. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Aos meus pais, Alberto e Mariana,
que empreenderam minha vida
com amor, sabedoria e suor.

Agradecimentos

Agradeço a minha esposa, Giulianna, meu eterno amor, pelo apoio e carinho nesses 12 anos, e sem dúvida, também pela paciência. Aos meus irmãos, Leonardo e Renato, eternos companheiros. E aos meus filhos, Nina e Theo, que iluminam diariamente minha vida e que, como esta tese, são frutos desta jornada de 4 anos.

Agradeço a minha tia, Lilian Nasser, que o sucesso acadêmico sempre me serviu de inspiração, e que sempre esteve disponível para ajudar. E ao meu tio, Leibo Kampela, cuja sabedoria sempre estará ao meu lado.

Agradeço ao meu orientador, Prof. Hélio Côrtes Vieira Lopes, que trabalha arduamente e com paixão para consolidar a PUC-Rio como uma referência mundial em *Data Science* e que soube me motivar e administrar. Espero sempre retribuir seu apoio com resultados. Agradeço também ao Prof. Marco Antonio Casanova, meu eterno orientador, por sempre me incentivar.

Agradeço ao Bruno Guberfain do Amaral, pela parceria em artigos e trabalhos nos últimos anos. E ao Gustavo Carvalho que possibilitou, com seu apoio profissional e pessoal, minha dedicação a este projeto.

Agradeço também a Microsoft pelo fornecimento de créditos para execução dos experimentos e a CAPES e ao CNPq pelo apoio aos meus estudos.

Por último, agradeço a PUC-Rio pelos auxílios concedidos e a todos os professores do Departamento de Informática, cujo convívio, ano a ano, torna-se cada vez mais gratificante.

Resumo

Nasser, Rafael Barbosa; Lopes, Hélio Côrtes; **Uma Plataforma na Nuvem para Armazenamento de Dados Georreferenciados de Mobilidade Urbana**. Rio de Janeiro, 2016, 159 p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A qualidade de vida nos grandes centros urbanos tem sido motivo de preocupação para governantes, empresários e para a população residente em geral. Os serviços de transporte público coletivo exercem papel central nessa discussão, uma vez que determinam, sobretudo para aquela camada da sociedade de menor poder aquisitivo, o tempo desperdiçado diariamente em seus deslocamentos. Nas metrópoles brasileiras, os ônibus municipais são predominantes no transporte coletivo. Os usuários deste serviço – passageiros – não dispõem de informações atualizadas sobre os ônibus e linhas de ônibus em operação. Oferecer essa natureza de informação contribui para uma melhor experiência de uso diário deste modal e, consequentemente, proporciona maior qualidade de vida aos seus usuários. Em uma visão mais abrangente, os ônibus podem ser considerados sensores que viabilizam a compreensão dos padrões e identificação de anomalias no tráfego de veículos nas áreas urbanas, possibilitando galgar benefícios para toda população. O presente trabalho apresenta uma **plataforma na nuvem** que **captura**, **enriquece**, **armazena** e **disponibiliza** os **dados** dos dispositivos de GPS instalados nos ônibus, permitindo a extração de conhecimento a partir deste valioso e volumoso conjunto de informações. Experimentos são realizados com os ônibus do Município do Rio de Janeiro, com aplicações focadas no passageiro e na sociedade. As metodologias, discussões e técnicas empregadas ao longo do trabalho poderão ser reutilizados para diferentes cidades, modais e perspectivas.

Palavras-chave

Informática; Engenharia de Software; Data Science; Computação na Nuvem; Computação Paralela; Geoprocessamento; Mobilidade Urbana.

Abstract

Nasser, Rafael Barbosa; Lopes, Hélio Côrtes (Advisor); **A Cloud Computing Platform for Storing Georeferenced Mobility Data**. Rio de Janeiro, 2016, 159 p. Doctoral Thesis – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The quality of life in urban centers has been a concern for governments, business and the resident population in general. Public transportation services perform a central role in this discussion, since they determine, especially for that layer of lower-income society, the time wasted daily in their movements. In Brazilian cities, city buses are predominant in public transportion. Users of this service - passengers - do not have updated information of buses and lines. Offer this kind of information contributes to a better everyday experience of this modal and therefore provides greater quality of life for its users. In a broader view, the bus can be considered sensors that enable the understanding of the patterns and identify anomalies in vehicle traffic in urban areas, allowing benefits for the whole population. This work presents a **platform in the cloud** computing environment that **captures**, **enriches**, **stores** and **makes available** the **data** from GPS devices installed on buses, allowing the extraction of knowledge from this valuable and voluminous set of information. Experiments are performed with the buses of the Municipality of Rio de Janeiro, with applications focused on passenger and society. The methodologies, discussions and techniques used throughout the work can be reused for different cities, modal and perspectives.

Palavras-chave

Computer Science; Software Engineering; Data Science; Cloud Computing; Parallel Computing; Geoprocessing; Urban Mobility.

Sumário

Lista de figuras

Lista de tabelas

1
Introdução

1.1
Motivação

O congestionamento no tráfego em áreas urbanas é com certeza um grande problema, principalmente porque ele causa um enorme prejuízo econômico. Vários esforços no mundo têm sido feitos para melhor controlar o tráfego e planejar a mobilidade em cidades [1]. Uma possível solução para diminuir os congestionamentos nas cidades é prover um sistema público de transporte eficiente, pois ele pode desempenhar um papel extremamente importante nas grandes cidades. Em [2], os autores mencionam que as principais razões desse importante papel são: redução do custo de transporte; ofertar um serviço para toda a sociedades e; economia de energia.

O sistema de transporte público na Cidade do Rio de Janeiro é historicamente deficiente, principalmente porque é baseado em um antigo sistema de ônibus. Para mudar esse estado, a Prefeitura deflagrou algumas iniciativas, tal como o desenvolvimento de um projeto de dados abertos que disponibiliza na web, a cada minuto, a posição instantânea de todos os ônibus na cidade. Apesar de não ser uma tecnologia nova, é uma primeira iniciativa a ser desenvolvida no Rio. Nesse trabalho constituímos um imenso conjunto de dados de posições de GPS (mais de 3 bilhões de entradas) de todos os ônibus que operaram na cidade desde Junho de 2014. Esses dados foram adquiridos no portal de dados abertos da Prefeitura [6]. Esse portal não armazena o histórico de dados, somente disponibiliza as posições correntes dos ônibus.

Podemos considerar que esses dados de GPS, que são transmitidos de minuto em minuto, representam sensores móveis de tráfego da Cidade. Mais ainda, as trajetórias desses ônibus podem ser entendidas como um *data stream* continuamente gerados.

Como a cidade do Rio de Janeiro possui uma rede densa de ônibus, as trajetórias adquiridas formam uma base de dados muito útil para análise de tráfego. De fato, essas trajetórias formam uma fonte estável de dados, pois os ônibus percorrem uma rota fixa nas ruas da cidade, em intervalos de tempo

regulares, quando as condições de tráfego permitem. Portanto, esses dados podem ser utilizados na detecção de anormalidades no trafego.

Este trabalho apresenta uma plataforma na nuvem que **captura**, **enriquece**, **armazena** e **disponibiliza** os dados dos dispositivos de GPS instalados nos ônibus, permitindo a extração de conhecimento a partir deste valioso e volumoso conjunto de dados. Em uma visão geral, essa plataforma:

• **Captura**, de forma automática, continuada e estável, os conjuntos de dados disponibilizados no Portal de Dados Abertos da Prefeitura do Rio de Janeiro sobre o serviço de ônibus do município da Cidade, em especial, as posição dos veículos, discretização dos trajetos das linhas de ônibus e seus pontos de parada, além de outras informações, como, por exemplo, as tarifas cobradas.

• **Enriquece** os dados históricos acumulados e os recém capturados, transformando, desta forma, dados em conhecimentos. A arquitetura implantada permite que novas estratégias, algoritmos e serviços sejam desenvolvidos, favorecendo assim a extração de novos conhecimentos quando desejado.

• **Armazena** esses conjuntos de dados e os conhecimentos extraídos de forma a racionalizar o uso de recursos computacionais e otimizar o tempo de sua recuperação buscando, assim, um bom equilíbrio entre essas perspectivas.

• **Disponibiliza** os dados, informações e conhecimentos neste âmbito, através de um conjunto de serviços, viabilizando o desenvolvimento de soluções clientes que os consumam. Tais soluções podem adotar distintas tecnologias, ambientes (e.g. web, desktop e móvel), objetivar o atendimento a diferentes públicos (e.g. usuário, população, governantes e empresários de ônibus) e, ainda, visar a extração de conhecimentos adicionais, possivelmente agregando outras fontes de dados (e.g. sua localização com um smartphone).

Em outras palavras, essa plataforma apoia a realização de atividades de Data Science, conforme detalhado nas seções que se seguem, i.e. a transformação dos dados brutos sobre os ônibus em conhecimentos relevantes. Cabendo ressaltar que as metodologias, discussões e técnicas empregadas ao longo do trabalho poderão ser reutilizados para diferentes cidades, modais e perspectivas.

Em função da sua aplicação inicial no modal de ônibus (“Bus”), na cidade do Rio de Janeiro (“Rio”) , a plataforma foi batizada “BusesInRio”, nome que utilizaremos ao longo do trabalho para identificar a plataforma em questão.

1.2
Data Science

A área de **Data Science** tem como objetivo transformar dados em informação e informação em conhecimento [7]. Se pauta no uso de métodos computacionais para fazer descobertas valiosas e práticas a partir de conjuntos de dados brutos. Para tanto, envolve uma ampla gama de atividades, tal como indicado pelo ciclo apresentado na Figura 1.

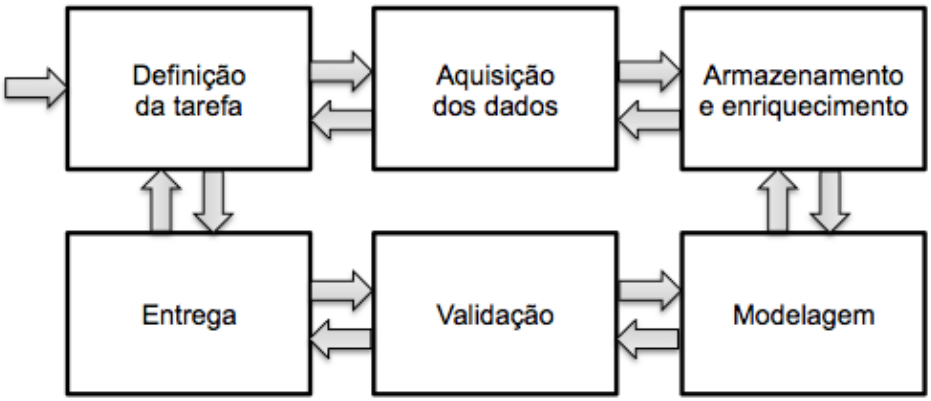


Figura 1 – Ciclo de atividades envolvidas em projetos de Data Science

Atualmente **Data Science** integra conhecimento gerado por várias outras áreas de conhecimento (Matemática, Probabilidade e Estatística, por exemplo). Em particular, na própria Ciência da Computação ela abrange diversas subáreas, que estão listadas em ordem alfabética: Aprendizado de Máquina, Banco de Dados, Computação de Alta Performance, Computação Gráfica, Engenharia de Software, Inteligência Artificial, Interação Humano-Computador, Otimização, dentre outras.

No universo onde os dados são intensamente gerados, eles se tornam um ativo crucial, e a área de Data Science está impulsionando novas pesquisas [8], novas formas de se fazer negócios [9] e uma nova formação acadêmica [10].

O principal objetivo de um projeto em **Data Science** consiste em otimizar processos e apoiar melhores tomadas de decisão informadas por **dados**, gerando um **aumento de valor a eles** [11].

O Instituto Gartner [12] resumiu num gráfico, apresentado na Figura 2, uma escala de valorização dos dados à medida que a sua análise se torna mais

sofisticada. Nessa figura, o Instituto Gartner lista quatro diferentes tipos de **análise de dados**:

1. **Análise Descritiva**: faz um exame dos dados para responder a pergunta "O que aconteceu?" (ou "O que está acontecendo?"). Usa muitos conceitos de Business Inteligence (BI) e de exploração visual dos dados.

2. **Análise Diagnóstica**: é o tipo de análise que examina os dados a fim de responder a pergunta "Porque isso aconteceu?" e é caracterizada pelo uso de técnicas tais como mineração dos dados e correlações.

3. **Análise Preditiva**: é um tipo de análise avançada dos dados que examina os dados para responder "O que mais possivelmente vai acontecer?". Essa análise é caracterizada pelo uso de técnicas de forecasting e de aprendizado de máquina.

4. **Análise Prescritiva**: é o tipo mais avançado de análise de dados cujo objetivo é examinar os dados a fim de responder "O que pode ser feito?" ou "O que podemos fazer para que isso aconteça?". É caracterizada por uso de técnicas tais como análise em grafos, simulação, processamento de eventos complexos, sistemas de recomendação, heurísticas e aprendizado de máquina.

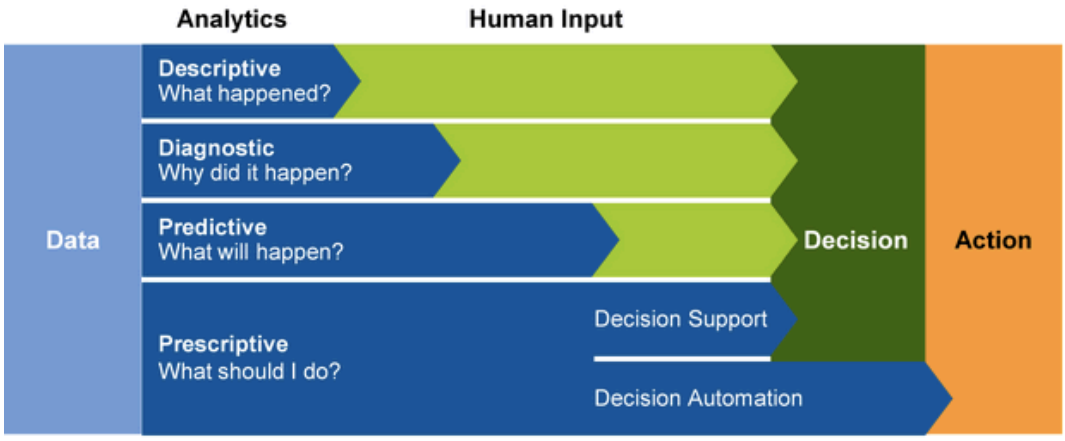


Figura 2 – Escala de valorização dos dados (Gartner 2016)

O presente trabalho não se limita ao desenvolvimento de uma plataforma na nuvem para extração de conhecimentos a partir de dados de localização georeferenciada de mobilidade urbana. Contemplando também a efetiva extração de um conjunto de conhecimentos, no campo das análises descritivas e

diagnósticas, visando suportar melhores tomadas de decisão a partir de dados, conforme detalhamos a seguir.

1.3 Conhecimentos Extraídos

A qualidade de vida nos grandes centros urbanos tem sido motivo de preocupação para governantes, empresários e para a população residente em geral. Os serviços de transporte público coletivo exercem papel central nessa discussão, uma vez que determinam, sobretudo para aquela camada da sociedade de menor poder aquisitivo, o tempo desperdiçado diariamente em seus deslocamentos. Nas metrópoles brasileiras, os ônibus municipais são predominantes no transporte coletivo.

Compreender melhor este serviço e os padrões de tráfego das principais vias urbanas pode certamente ajudar na tomada de decisões melhores, que vão da simples escolha do melhor horário para se sair de casa ao complexo planejamento deste serviço.

Em particular, os usuários deste serviço – passageiros – não possuem informações atualizadas quanto ao trajeto das linhas de ônibus, não dispõem de uma tabela de horários para planejarem suas viagens e muito menos conhecem o tempo estimado de chegada em seu destino final. Oferecer essa natureza de informação contribui para uma melhor experiência de uso diário deste modal e, consequentemente, proporciona maior qualidade de vida aos seus usuários.

O presente trabalho apresenta também o uso da Plataforma BusesInRio desenvolvida na transformação desse imenso conjunto de dados - posições de GPS (mais de 3 bilhões de entradas) de todos os ônibus que operaram na cidade do Rio de Janeiro desde Junho de 2014 e demais dados disponibilizados deste serviço - em conhecimentos, visando:

- Disponibilizar informações aos usuários sobre o serviço em operação;
- Dar transparência à população das características históricas do serviço;
- Descobrir os padrões de tráfego nas principais vias da Cidade;
- Identificar a situação do tráfego das principais vias da Cidade.

Certamente o conjunto de conhecimentos extraído é apenas uma pequena amostra do potencial destes dados e dos benefícios da **Data Science**, mas permitem validar o uso Plataforma BusesInRio em cenários reais e contribuir com uma melhor compreensão do potencial destes dados. Salientamos que o foco da realização destes trabalhos é destacar a abrangência de situações em que essa plataforma pode ser empregada e validar sua utilização de forma completa, da captura a entrega do conhecimento.

1.4 Trabalhos Relacionados

O presente trabalho teve como inspiração inicial [13], que apresentou diversos gráficos para representar o comportamento dos ônibus ao longo de suas linhas, a partir dos dados disponibilizados pela Prefeitura do Rio de Janeiro em relação aos ônibus municipais da Cidade. Essas visualizações se restringiram ao comportamento de algumas linhas de ônibus, com rotas previamente conhecidas. Tais visualizações foram processadas individualmente em máquina local, a partir de subconjuntos pré-selecionados dos dados. Em [14] outras análises e visualizações são propostas a partir de um pequeno subconjunto destes mesmos dados, também fortemente amparadas nas linhas de ônibus.

Extrapolando o universo dos ônibus e suas linhas, existem diversas pesquisas e sistemas para monitorar e visualizar trajetórias (i.e. sequencia ordenada de pares de informação temporal e espacial, tais como *timestamp* e latitude/longitude). São exemplos: [15] que utiliza dados de trajetórias de táxis para gerar relatórios de monitoramento do tráfego da cidade, [16] e [17], onde são apresentas soluções *web* para visualizar e analisar as condições do tráfego e suas tendências, [18] que estima o estado do tráfego a partir de sensores em carros, [19] que propõe uma visualização em múltiplos níveis, além de [20], [21] e [22].

De igual forma, muitos trabalhos fazem uso de outras fontes de dados para enriquecer a análise destas trajetórias, tais como [23] e [24] que descrevem um sistema de monitoramento de frotas de caminhões usando o Twitter e [25] que utiliza vídeos das autoestradas.

Do ponto de vista de tempo estimado de chegada, também é possível localizar muitos trabalhos na literatura, sendo [26] apenas um exemplo. Também podemos exemplificar um trabalho relacionado a tabelas de horários, a citar [27].

A literatura em relação a detecção e avaliação de anomalias no tráfego é muito rica, cabendo destaque para [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39] e [40]. Outra perspectiva interessante é a clusterização de trajetórias, tratadas, por exemplo, em [41], [42], [43], [44] e [45]. Por último, outros estudos relacionados são igualmente interessantes, a saber: [46], [47] e [48].

Merece especial atenção [84], onde é apresentado um sistema capaz de realizar consultas e mineração de dados a partir de dados de mobilidade, tendo trajetórias como seu conceito central. Sem dúvida essa ferramenta permite realizar visualizações, estatísticas e aplicar algoritmos pré-existentes com enorme utilidade, no entanto, como esperado, por ser um sistema, está habilitado a suportar funcionalidades especificas pré-concebidas pelos seus desenvolvedores.

Diferente dos trabalhos mencionados, este objetiva a construção de um alicerce sólido – a Plataforma BusesInRio – para que conhecimentos sejam extraídos a partir do conjunto integral de dados capturados e não apenas um subconjunto. Não se limitando a algumas experimentações, como na maioria dos trabalhos citados, permitindo que serviços sejam instanciados de forma a extrair continuamente conhecimentos relevantes dos dados ou novas implementações sejam incluídas a partir do ambiente pré-existente. Também se diferencia pelo uso de serviços na nuvem e seu potencial de processamento distribuído, como explorado em [79].

Nesse trabalho, espera-se também instrumentar este processo com algoritmos relevantes, para que fique demonstrada a abrangência desta Plataforma BusesInRio, que é capaz de endereçar abordagens distintas sobre dados georreferenciados.

Especificamente, utilizamos dados da Cidade do Rio de Janeiro e desenvolvemos formas de visualização e consumo destes conhecimentos extraídos, para que o trabalho cubra do início ao fim o processo de transformação de dados em conhecimento, no entanto, como mencionado anteriormente, outros modais, localidades e algoritmos podem ser utilizados na Plataforma BusesInRio.

1.5 Organização do Trabalho

O presente trabalho se encontra estruturado em 8 capítulos, a saber:

• Introdução **(Capítulo 1):** Apresenta de forma concisa a motivação, objetivos e organização do trabalho, assim como, os trabalhos relacionados.

• Compreendendo o Domínio e os Dados Disponíveis **(Capítulo 2):** Discorre sobre a mobilidade urbana, o transporte público coletivo e os ônibus municipais do Rio de Janeiro, fornecendo informações relevantes sobre os mesmos. Resume a Lei de Acesso à Informação e também o conceito de dados abertos, assim como sua aplicação neste domínio. Por último, detalhamos os dados disponíveis sobre os ônibus municipais da Cidade.

• Coleta, Armazenamento e Visualização Básica dos Dados **(Capítulo 3):** Apresenta o processo de coleta de dados, assim como a forma com que os mesmos são armazenados. Utilizando um Web Site desenvolvido são realizadas as primeiras visualizações básicas dos dados, para melhor compreendê-los.

• A Plataforma **(Capítulo 4):** Contextualiza o objetivo da Plataforma BusesInRio e o conceito de computação em nuvem, justificando seu uso. A sua arquitetura é detalhada, sendo explicado seus serviços e métodos de comunicação disponíveis, assim como, são apresentadas de forma geral as soluções clientes da Plataforma BusesInRio, desenvolvidos no âmbito deste trabalho.

• Extraindo Conhecimento dos Dados sobre o Tráfego **(Capítulo 5):** Esse capítulo explora com profundidade a descoberta de conhecimento a partir dos dados, detalhando como cada conhecimento é extraído. A seleção de segmentos, a extração de trajetórias e detecção da situação de um ônibus são alguns exemplos dos algoritmos apresentados.

• Tomando Decisão com Dados **(Capítulo 6):** Os serviços disponibilizados pela Plataforma BusesInRio são explorados em uma aplicação para dispositivos móveis. Esta oferece funcionalidades à população e aos usuários do serviço de ônibus, que agregam características do próprio dispositivo, evidenciando assim, a multiplicidade de usos da Plataforma.

• Conclusão **(Capítulo 7):** Neste capítulo destacamos as contribuições, dificuldades, lições aprendidas, limitações e perspectivas futuras do trabalho.

• Referências **(Capítulo 8):** Coletânea de artigos , livros e sites relevantes aos assuntos tratados no âmbito deste trabalho.

2
Compreendendo o Domínio e os Dados Disponíveis

2.1
Mobilidade Urbana e Transporte Público Coletivo

A população do Brasil vive 84,4% em áreas urbanas, segundo o censo demográfico de 2010 do Instituto Brasileiro de Geografia e Estatística [49].

Dentro deste espaço, os serviços de transporte público coletivo exercem papel fundamental, uma vez que determinam, sobretudo para as classes desfavorecidas, o grau de praticidade dos seus deslocamentos diários. Pesquisa do Instituo de Pesquisa Econômica e Aplicada [50] indica que o transporte público coletivo é o meio de transporte mais utilizado pelos brasileiros dentro da cidade (44,3%), seguido pelo carro (23,8%), moto (12,6%), a pé (12,3%) e, por último, bicicleta (7,0%).

De acordo com a Constituição Federal de 1998 [51], compete aos Municípios organizar e prestar, diretamente ou sob regime de concessão ou permissão, o serviço de transporte público coletivo. A Lei Federal 12.581 de 2 de janeiro de 2012 [52], institui as diretrizes da política nacional de mobilidade urbana e define, no seu art. 4º, a mobilidade urbana como *"condição em que se realizam os deslocamentos de pessoas e cargas no espaço urbano"*. Esse mesmo artigo define transporte público coletivo como *"serviço público de transporte de passageiros acessível a toda a população mediante pagamento individualizado, com itinerários e preços fixados pelo poder público"*.

O Município deve prover condições de locomoção de maneira que seus habitantes possam exercer seu direito de ir e vir livremente, de forma rápida e eficiente. Para tanto, deve ser ofertada infraestrutura e demais elementos para essa movimentação, com transporte público viário, ferroviário e fluvial com uma sistemática inteligente. Adicionalmente, o transporte individual por meio de automóveis ou veículos movidos à tração humana também devem ser facilitados pelas autoridades urbanas neste perímetro.

Para a obtenção de um serviço de qualidade é fundamental a conciliação de interesses de três grupos com preocupações distintas quanto ao desempenho de um sistema de transporte público, sendo eles:

- **Usuários:** utilizam o serviço público para suprir suas necessidades de deslocamento e não tem maiores preocupações com a operação dos serviços. Na tomada de decisão quanto ao uso do transporte público, levam em consideração a regularidade, tempo de deslocamento, conforto, custos, entre outros atributos.
- **Operadores:** encarregados de administrar, operar (financiamento, aquisição, manutenção, renovação da frota, etc.) e comercializar o transporte coletivo, sob a forma de prestação de um serviço público. Suas preocupações estão relacionadas com as variáveis que influenciam os custos e receitas na oferta do serviço.
- **Poder público:** responsável legal pelo transporte público; deve regulamentar, planejar, programar e fiscalizar a execução dos serviços, interagindo nos conflitos de interesse existentes entre usuários e operadores, através de legislação específica. Fica, também, a cargo do Poder Público defender os interesses de um quarto grupo, a comunidade em geral, cujo interesse é indireto, provocado pelas externalidades do sistema de transporte publico (ruído excessivo, poluição ambiental, conflitos com o uso do solo, etc.).

Um cenário ideal, com a integração de transporte coletivo e transporte individual em harmonia com a cidade, sem atrasos e completamente acessível para toda a população é o sonho de muitos dos cidadãos que chegam a passar 3 horas por dia, ou mais, fazendo o trajeto casa-trabalho-casa. Algumas cidades do mundo, que ainda estão longe deste cenário utópico, já passaram por mudanças radicais e se aproximam de alguns objetivos, que representam uma grande melhora na qualidade de vida de seus moradores. Tais metamorfoses, que se dão de médio a longo prazo, podem um dia trazer uma efetiva melhoria no dia a dia das pessoas.

Em particular, no Brasil, a pesquisa "Mobilidade Urbana e Cidadania" da Diretoria de Análise de Políticas Públicas [53] da Fundação Getúlio Vargas (FGV), realizada nas regiões metropolitanas de São Paulo, Rio de Janeiro, Brasília, Belo Horizonte, Recife e Porto Alegre, entre 25 de março e 07 de abril de 2014, apontou que mais 70% dos usuários do transporte público estão "insatisfeitos" com o serviço. Assim como, para 65% dos entrevistados o transporte público não melhorou nos últimos anos e, ainda, para 60% dos entrevistados, não existe crença que vai melhorar nos próximos.

O Município do Rio de Janeiro, conforme dados de abril de 2015 [54], tem a participação dominante do modal "Ônibus Municipal" no transporte público coletivo, sem alterações previstas em curto prazo, conforme detalhado na Figura 3.

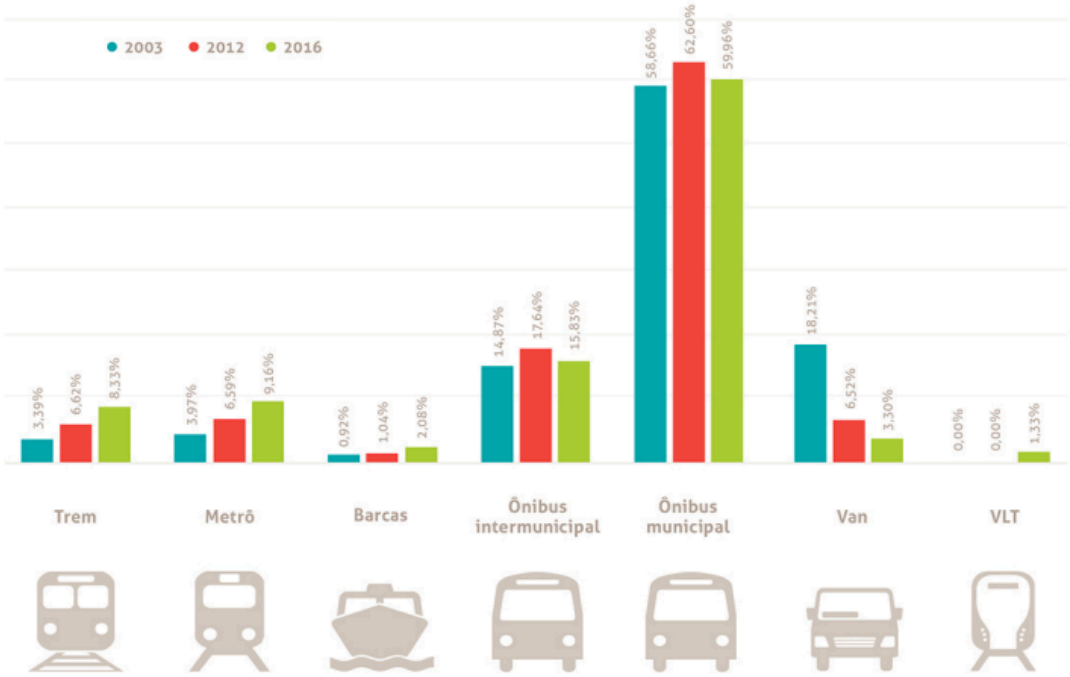


Figura 3 – Evolução dos Principais Modais do Transporte Coletivo

Diante do exposto, buscar formas de melhorar a experiência do usuário de transporte público, em especial, de Ônibus Municipal, possivelmente represente uma melhoraria na qualidade de vida da população.

2.2
Ônibus Municipal do Rio de Janeiro

As estatísticas da Fetranspor (Federação das Empresas de Transportes de Passageiros do Estado do Rio de Janeiro) [55], referentes ao ano de 2015 e ao município do Rio de Janeiro, indicam números expressivos deste serviço:

- 705 linhas de ônibus em operação na época;
- 9.008 ônibus em operando na época;
- 19.075.362 viagens realizadas;
- 1.085.326.478 passageiros;
- 723.478.360 quilômetros percorridos;
- 43 empresas com 39.596 empregados;

- 4,38 anos de idade média da frota;
- 1,5 é índice de passageiros por quilômetro; e
- 6.694 km de média mensal percorrida por ônibus.

Observe que segundo o número acima temos o equivalente a 47,05% da população carioca - que é na ordem de 6.320.446 [56] - utilizando este serviço diariamente. Portanto, melhorar este serviço ou pelo menos a experiência de uso do mesmo representa influenciar de forma positiva e significativa a qualidade de vida do morador do Município do Rio de Janeiro.

Este modal, "Ônibus Municipal" de transporte público coletivo no Município do Rio de Janeiro é operado por um consórcio de empresas, que se divide em quatro áreas, conforme representado na Figura 4.

Figura 4 – Divisão Territorial do RJ pelos Consórcios de Ônibus

2.3
Lei de Acesso à Informação

Acesso às informações sob a guarda de órgãos e entidades públicas é direito fundamental do cidadão e dever do Estado, conforme previsto pela Constituição brasileira e regulamentado pela Lei Federal 12.527, sancionada em 18 de novembro de 2011 pela Presidenta da República.

Esse direito fundamental aparece expresso no artigo 5º da Constituição: *"todos são iguais perante a lei, sem distinção de qualquer natureza, garantindo-se aos brasileiros e aos estrangeiros residentes no País a inviolabilidade do direito à vida, à liberdade, à igualdade, à segurança e à propriedade"*. No inciso XXXIII deste artigo, determina-se que *"todos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou*

geral, que serão prestadas no prazo da lei, sob pena de responsabilidade, ressalvadas aquelas cujo sigilo seja imprescindível à segurança da sociedade e do Estado".

No artigo 37 da Constituição Federal fica explícito que *"a administração pública direta e indireta de qualquer dos Poderes da União, dos estados, do distrito federal e dos municípios obedecerá aos princípios de legalidade, impessoalidade, moralidade, publicidade e eficiência".*

Em função dos avanços tecnológicos, novas questões passam a se relacionar com esses princípios: Como garantir o direito de receber informações públicas quando a relação entre governo e cidadãos é intermediada pelas máquinas? O que significa ser público e eficiente em tempos de Internet acessível a todos?

Nesse contexto, a Lei 12.527 de 18 de novembro de 2011, Lei de Acesso à Informação, representa uma mudança de paradigma em matéria de transparência pública, pois estabelece que o acesso é a regra e o sigilo a exceção. Qualquer cidadão poderá solicitar acesso às informações públicas, ou seja, aquelas não classificadas como sigilosas, conforme procedimento que deverá observar as regras, prazos, instrumentos de controle e recursos previstos.

O desafio, agora, é assegurar a implementação desta Lei, que apresenta diversos desafios de natureza técnica e tecnológica e, também, de caráter administrativo, que incluem a necessidade de recursos financeiros e humanos - estes, devidamente capacitados - para garantir a observância do que dispõe a Lei. Além disso, não menos importante, é transpor a cultura do sigilo que, de forma silenciosa e invisível, ainda se constitui um dos grandes obstáculos para a abertura dos governos.

A Controladoria-Geral da União disponibilizou uma cartilha onde maiores informações podem ser obtidas sobre a lei, os aspectos e vantagens de uma cultura administrativa pró-acesso à informação.

2.4
Dados Abertos

"Dado aberto é um dado que pode ser livremente utilizado, reutilizado e redistribuído por qualquer um" (Site Dados Abertos). A definição completa fornece detalhes específicos do significado do termo, abaixo sumarizados.

- **Disponibilidade e acesso:** o dado precisa estar disponível por inteiro e por um custo razoável de reprodução, preferencialmente, por meio de *download* na Internet; também deve estar em um formato conveniente e modificável.
- **Reuso e redistribuição:** o dado precisa ser fornecido em condições que permitam reutilização e redistribuição, incluindo o cruzamento com outros conjuntos de dados.
- **Participação universal:** todos podem usar, reutilizar e redistribuir, não havendo discriminação contra áreas de atuação, pessoas ou grupos, por exemplo, não são permitidas restrições como "não comercial", que impedem o uso comercial e restrições de uso para determinados fins, como "somente educacional".

Corroborando com esta definição é enriquecedor elencar os princípios pregados pelos ativistas de dados abertos, a saber.

- **Acessíveis:** são disponibilizados para a maior quantidade possível de pessoas, atendendo, assim, aos mais diferentes propósitos.
- **Completos:** todos os dados públicos devem ser disponibilizados. Dado público é aquele que não está sujeito à restrições de privacidade, segurança ou outros privilégios.
- **Primários:** são apresentados tal como colhidos da fonte, com o maior nível possível de granularidade, sem agregação ou modificação. Por exemplo, um gráfico não é fornecido aberto mas, sim, os dados utilizados para construir a planilha que deu origem a ele podem ser abertos.
- **Atuais:** devem ser publicados o mais rápido possível para preservar seu valor. Em geral têm periodicidade, quanto mais recentes e atuais, mais úteis para seus usuários.
- **Compreensíveis por máquina:** devem estar estruturados de modo razoável, possibilitando que sejam processados automaticamente, por exemplo, uma tabela em PDF é muito bem compreendida por pessoas, mas para um computador é apenas uma imagem; uma tabela em formato estruturado, como CSV ou XML, é processada mais facilmente por *softwares*.

- **Não discriminatórios:** Devem estar disponíveis para qualquer pessoa, sem necessidade de cadastro ou qualquer outro procedimento que impeça o acesso.
- **Não proprietários:** Nenhuma entidade ou organização deve ter controle exclusivo sobre os dados disponibilizados.
- **Livres de licenças:** Não devem estar submetidos a *copyrights*, patentes, marcas registradas ou regulações de segredo industrial. Restrições razoáveis quanto a privacidade, segurança e outros privilégios são aceitas, desde que transparentes e bem justificadas.

Dados abertos governamentais são dados produzidos pelo governo e colocados à disposição das pessoas de forma a tornar possível não apenas sua leitura e acompanhamento, mas também sua reutilização em novos projetos, sites e aplicativos; seu cruzamento com outros dados de diferentes fontes; e sua disposição em visualizações interessantes e esclarecedoras. No processo de abertura de dados, o foco não está nos dados pessoais, mas naqueles que não contêm informação sobre indivíduos específicos.

A interoperabilidade i.e. a capacidade de diversos sistemas e organizações trabalharem juntos, sendo esta a ideia central desta proposta.

Dados abertos, especialmente os governamentais, são um ótimo recurso ainda muito pouco explorado, considerando que muitos indivíduos e organizações coletam uma gama de diferentes tipos de dados para executar suas tarefas. O governo é especialmente importante nesse contexto, tanto pela quantidade e centralidade dos dados que possui, quanto pelo fato de que tais dados serem essencialmente públicos, sendo esta, repise-se, uma das garantias fundamentais prevista pela Constituição Federal.

O debate sobre dados governamentais abertos ganhou a mídia em 2009. Em governos de vários países, tais como Estados Unidos, Reino Unido, Canadá e Nova Zelândia. Foram anunciandas iniciativas voltadas à abertura de dados públicos, e desde então, a informação aberta se propaga no mundo e sua utilização em prol da sociedade avança exponencialmente.

O Laboratório Brasileiro de Cultura Digital e o Núcleo de Informação e Coordenação do Ponto BR (NIC.br) publicaram um manual para governos que querem abrir dados, podendo ser usado por qualquer pessoa que queira saber mais sobre os aspectos técnicos, sociais e políticos dos referidos dados.

2.5
Dados Abertos no Município do Rio de Janeiro

O Portal de Dados Abertos da Prefeitura do Rio de Janeiro [6] é a ferramenta disponibilizada pelo Município para que a sociedade possa encontrar e utilizar os dados abertos governamentais da cidade do Rio de Janeiro, o que proporciona ao cidadão um melhor entendimento da Administração Pública, no que se refere ao acesso aos serviços públicos, controle das contas públicas e participação no planejamento, bem como melhor conhecimento da cidade. O Portal também tem o objetivo de promover a interlocução entre atores da sociedade e o governo para pensar a melhor utilização dos dados em benefício da sociedade.

Este canal funcionará como um grande catálogo que facilitará a busca, uso e reuso de dados publicados pela Prefeitura do Rio de Janeiro e disponibiliza o acesso às informações, entre outras, do modal rodoviário municipal fornecido pelos GPS instalados nos ônibus que circulam na cidade do Rio de Janeiro, saúde, educação e Disque-Rio (1746).

Em especial, quando falamos de Transporte e Mobilidade, as informações disponíveis no Portal são complementadas pelo site Transparência da Mobilidade, da Secretaria Municipal de Transportes (SMTP), onde estão disponibilizados todos os dados relativos às operações do serviço de ônibus como custos, receitas e indicadores de qualidade do serviço. Neste site estão os dados relativos à operação do Sistema de Transporte Público por Ônibus licitado.

2.6
Dados Abertos sobre os Ônibus do Município do Rio de Janeiro

Estão disponíveis as informações a seguir sobre os Ônibus Municipais do Rio de Janeiro no Portal anteriormente apresentado. A especificação de cada linha destes conjuntos de dados também é apresentada a seguir.

- **Posição dos ônibus:** Conjunto de dados com a posição - coordenadas geográficas de latitude e longitude do sistema global de posicionamento (GPS) - instantânea de cada veículo. As coletas, a priori, tem intervalo de 1 minuto entre elas e os dados não são armazenados para consulta ao histórico. Além disso, os dados são disponibilizados em dois arquivos no formato JSON, um para linhas convencionais e outro para linhas do

sistema BRT, incluindo, neste segundo, as linhas alimentadores do sistema. O primeiro passou a ser disponibilizado em Março de 2014 e o segundo em Julho de 2015. Posteriormente, uma versão mais abrangente do primeiro foi também disponibilizada. A Tabela 1 lista a descrição dos campos disponibilizados.

Campo	Descrição
DataHora	Data e hora da coleta do dado
Ordem	Identificação alfanumérica encontrada na lateral dos ônibus
Linha	Linha do ônibus
Latitude	Latitude do ônibus na coleta (GPS, WGS84)
Longitude	Longitude do ônibus na coleta (GPS, WGS84)
Velocidade	Velocidade do ônibus na hora da coleta do dado

Tabela 1 – Posição dos ônibus

- **Pontos de parada das linhas do ônibus:** Conjunto de dados com a posição conhecida de todos os pontos de parada das linhas de ônibus. O ponto de ônibus ou ponto de parada são locais designados para um ônibus parar temporariamente para os passageiros embarcarem ou desembarcarem. Este dado, disponibilizado em formato CSV, sendo um arquivo por linha, a priori, é atualizado quando ocorrem mudanças nas linhas de ônibus. A Tabela 2 lista a descrição dos campos disponibilizados.

Campo	Descrição
Linha	Identificador da Linha de Ônibus
Descrição	Descrição da Linha de Ônibus
Sequência	Posição da Parada de Ônibus iniciando em zero
Latitude	Latitude da Parada de Ônibus (GPS, WGS84)
Longitude	Longitude do Parada de Ônibus (GPS, WGS84)

Tabela 2 – Pontos de parada das linhas do ônibus

- **Coordenadas dos trajetos das linhas de ônibus:** Conjunto de dados que discretiza os trajetos das linhas de ônibus, onde cada descrição representa uma posição. Uma linha de ônibus pode ser entendida como um serviço de transporte público regular que percorre determinado itinerário. Por sua vez, o itinerário é a definição dos trajetos a serem percorridos pelos ônibus desta linha. Este dado, disponibilizado em

formato CSV, sendo um arquivo por linha, a priori, é atualizado quando ocorrem mudanças nas linhas de ônibus. A Tabela 3 lista a descrição dos campos disponibilizados.

Campo	Descrição
Linha	Identificador da Linha de Ônibus
Descrição	Descrição da Linha de Ônibus
Sequência	Posição do Ponto iniciando em zero (referente ao "shape")
Shape_id	Identificador de uma Trajetória realizada pela Linha ("shape")
Latitude	Latitude do Ponto (GPS, WGS84)
Longitude	Longitude do Ponto (GPS, WGS84)

Tabela 3 – Coordenadas dos trajetos das linhas de ônibus

- ***General Transit Feed Specification*** **(GTFS)** [57]**:**

  Formato comum para horários de transportes públicos e de informação geográfica associada, proposto pelo Google, para agências e desenvolvedores interoperarem. Teoricamente atualizado quando ocorrem mudanças nas linhas de ônibus, pontos de parada e outras características do serviço e contêm vários arquivos em formato texto, conforme listado abaixo.

Arquivo	Definição
agency.txt*	Agência de transporte que provêm estas informações.
stops.txt*	Locais onde os veículos embarcam e desembarcam passageiros.
routes.txt*	Agrupamento de viagens em rotas.
trips.txt*	Viagens de cada rota (sequencia de pontos no tempo).
stop_times.txt*	Horário de chegada e partida de um veículo em determinado ponto para uma viagem.
calendar.txt*	Especifica quando um serviço inicia e quando termina.
calendar_dates.txt	Datas de exceção em relação ao calendário.
fare_attributes.txt	Tarifas.
fare_rules.txt	Regras para aplicação de tarifas por rota.
shapes.txt	Pontos do trajeto de uma rota.
frequencies.txt	Frequência entre rotas.
transfers.txt	Regras para conexão entre pontos de transferência das rotas.
feed_info.txt	Informação adicional sobre esse conjunto de dados.

Tabela 4 – Arquivos do GTFS

Em especial, no tocante aos dados do Rio de Janeiro, apenas os arquivos routes.txt, shapes.txt, trips.txt, fare_attributs.txt e fare_rules.txt são efetivamente relevantes. Os demais possuem informações depreciadas. Os dados contidos em cada arquivo e o relacionamento entre os mesmos é ilustrado no diagrama apresentado a seguir. Maiores detalhes podem ser obtidos em Google Transit APIs. A Tabela 4 lista a definição dos arquivos.

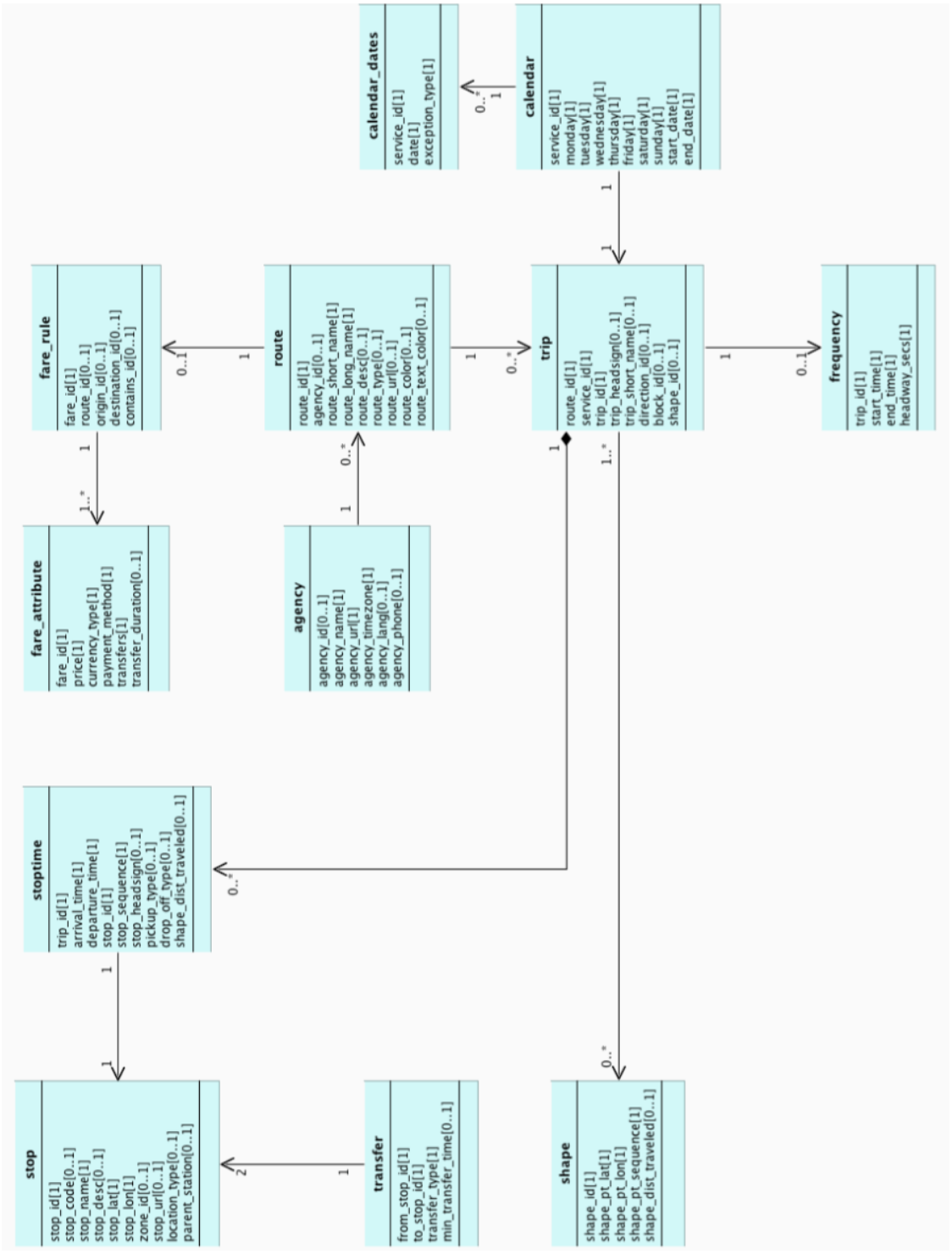


Figura 5 – Diagrama do GTFS

Mês	Tamanho	Registros	Sem Linha
Junho de 2014	1,01 GB	73.145.211	22,26%
Julho de 2014	1,49 GB	111.365.317	21,64%
Agosto de 2014	0,52 GB	38.491.314	20,80%
Setembro de 2014	2,00 GB	150.661.127	20,01%
Outubro de 2014	1,58 GB	120.618.942	19,82%
Novembro de 2014	1,12 GB	84.918.283	22,79%
Dezembro de 2014	1,40 GB	106.430.699	19,38%
2014	**9,12 GB**	**685.630.893**	**20,77%**
Janeiro de 2015	1,86 GB	143.626.826	24,74%
Fevereiro de 2015	1,79 GB	137.925.035	26,51%
Março de 2015	2,10 GB	161.585.524	23,26%
Abril de 2015	1,94 GB	149.803.857	23,19%
Maio de 2015	2,04 GB	156.447.803	20,15%
Junho de 2015	1,83 GB	140.238.082	20,97%
Julho de 2015	1,84 GB	140.770.355	20,94%
Agosto de 2015	1,45 GB	112.364.124	24,06%
Setembro de 2015	1,75 GB	133.602.361	24,82%
Outubro de 2015	2,16 GB	163.624.729	28,79%
Novembro de 2015	0,45 GB	33.764.704	26,41%
Dezembro de 2015	2,06 GB	155.356.670	25,59%
2015	**21,27 GB**	**1.629.110.070**	**24,12%**
Janeiro de 2016	2,21 GB	164.474.333	24,46%
Fevereiro de 2016	1,93 GB	142.424.654	21,65%
Março de 2016	2,00 GB	146.799.798	21,12%
Abril de 2016	1,75 GB	128.724.763	20,98%
Maio de 2016	1,58 GB	116.684.681	19,25%
Junho de 2016	1,89 GB	139.461.416	20,00%
Julho de 2016	1,87 GB	139.786.136	20,62%
Agosto de 2016	2,29 GB	174.424.424	23,83%
2016	**15,52 GB**	**1.152.780.205**	**21,49%**
Total	**45,91**	**3.467.521.168**	**22,19%**

Tabela 5 – Visão geral do histórico da posição dos ônibus

Esses conjuntos de dados são gerados pela Federação das Empresas de Transportes de Passageiros do Estado do Rio de Janeiro (Fetranspor) e mantidos pelo IPLAM (Empresa Municipal de Informática da Cidade do Rio de Janeiro).

Os conjuntos de dados **Pontos de parada das linhas do ônibus**, **Coordenadas dos trajetos das linhas de ônibus** e **GTFS** (veja a Figura 5) inicialmente eram coletados e utilizados apenas em sua versão atual, sem preocupação com a manutenção do histórico de atualizações. A partir de julho de 2016 iniciou-se o armazenamento diário de uma versão destes conjuntos de dados. Por outro lado, a partir de 12/06/2014 a **Posição dos ônibus** passou a ser capturada e armazenada, formando assim um histórico de posições destes ônibus, sendo acumulados mais de **3 bilhões** registros. Uma visão geral do histórico acumulado até 31/08/2016 é apresentada na Tabela 5.

O conjunto de dados do período de 12/06/2014 até 12/06/2015 foi gentilmente cedido por Bruno Amaral [13]. Estes foram uniformizados para que não apresentem qualquer diferença estrutural em relação aos posteriormente coletados, ou seja, foram tabulados utilizando os mesmos marcadores para quebra de linha e de campo e, também, compactados no mesmo formato (ZIP).

A Tabela 5 apresentou os dados capturados e armazenados a cada mês, com a respectiva totalização e subtotais por ano, sendo a temporalidade apresentada na sua primeira coluna.

Ônibus em circulação nestas datas	8.133
Linhas em circulação nestas datas	467
Linhas com paradas disponibilizadas	146 (31,26%)
Linhas sem paradas disponibilizadas	**321 (68,74%)**
Pontos de paradas disponibilizados	22.246
Média de pontos de paradas por linha	**152,37**
Linhas com trajetos disponibilizados	358 (76,66%)
Linhas sem trajetos disponibilizados	**109 (23,34%)**
Quantidade de trajetos disponibilizados	850
Média de trajetos por linha	**2,3743**
Pontos discretizando os trajetos disponibilizados	620.889
Média de pontos discretizados por trajeto	**730,45**

Tabela 6 – Visão geral dos Pontos de Parada e Trajetos dos ônibus

O campo “Tamanho” em *Gygabytes* (GB) corresponde ao espaço em disco necessário para armazenar de forma compactada esse conjunto de dados, enquanto o campo “Registros” apresenta o total de registros acumulado em cada período, onde cada registro corresponde a especificação apresentada na Tabela 1. Por último, é destacado no campo “Sem Linha” o percentual de registros que se encontram com a informação da linha do ônibus em branco, ou seja, que não possui a informação de qual linha pertence o ônibus cuja posição foi disponibilizada. Vale ressaltar que quase 22% dos dados do histórico não continham a relação entre ônibus e linha de ônibus, o que pode ocorrer pelo fato do ônibus estar fora de serviço ou por inadimplemento da operadora da linha.

Buscando ainda o entendimento desses conjuntos de dados, nos dias 15, 16 e 17/07/2015, foram obtidos os dados dos **Pontos de parada das linhas do ônibus** e dos **Pontos dos trajetos das linhas de ônibus** disponibilizados no Portal. Essas informações foram confrontadas com a **Posição dos ônibus** obtidas em 18/07/2015 também no Portal. As principais características identificadas são apontadas na Tabela 6.

O número de linhas cujos pontos de parada são desconhecidos é expressivo, sendo superior a 68%. Da mesma forma, o número de linhas cujos trajetos são desconhecidos, superior a 23%, é considerável. Por outro lado, o número de pontos de parada por linha se mostra exacerbado. Esses números indicam que os operadores do Serviço Público de Passageiros por Ônibus do Município do Rio de Janeiro (SPPO-RJ) não estão disponibilizando informações atualizadas e consistentes em relação a suas linhas de ônibus.

Por outro lado, o número médio de trajetos por linha é consistente com a ideia de um trajeto de ida e outro de volta, com baixíssimas variações divulgadas. No entanto, a maioria das linhas não sofreu atualização há algum tempo, o que gera a desconfiança de que muitas estejam desatualizadas, em especial, as que sabidamente passam em áreas que sofreram intervenções recentes, como, por exemplo, o Centro da Cidade.

Cabe registrar que no site Transparência da Mobilidade é possível encontrar outras informações sobre o SPPO-RJ, a saber:

- Edital de licitação do sistema de ônibus e contratos de concessão dos ônibus com a Transcarioca, Santa Cruz, Internorte e Intersul;

- Decreto Nº 38.279 de 29 de janeiro de 2014, que estabelece medidas para o aperfeiçoamento da prestação deste serviço;
- Gratuidade no transporte público para estudantes das redes públicas;
- Histórico do serviço e dos reajuste;
- Relatório de operações por linha;
- Indicadores de demanda, participação dos consórcios e de mercado;
- Ranking negativo (fiscalização direcionada sobre linhas de ônibus);
- Relatório diário de operações do BRT Transcarioca e BRT Transoeste;
- Relação dos postos de venda do bilhete único carioca.

3
Coleta, Armazenamento e Visualização Básica dos Dados

3.1
Dados Armazenados

O procedimento de coleta, que será detalhado na seção 3.2, armazena os arquivos abaixo.

- **Últimas 5 posições conhecidas de cada ônibus ("now.txt"):** Arquivo em formato texto que contêm em cada linha a informação das últimas cinco posições conhecidas de um ônibus em particular, conforme dados especificado na Tabela 7.

Campo	Descrição
DataHora	Data e hora da última coleta de dados do GPS do ônibus
Ordem	Identificação alfanumérica encontrada na lateral dos ônibus
Linha	Identificador da linha do ônibus
Latitude	Última latitude conhecida do ônibus (GPS, WGS84)
Longitude	Última longitude conhecida do ônibus (GPS, WGS84)
Velocidade	Última velocidade conhecida do ônibus na hora da coleta
DataHora1	Data e hora da penúltima coleta de dados do GPS do ônibus
Latitude1	Penúltima latitude conhecida do ônibus (GPS, WGS84)
Longitude1	Penúltima longitude conhecida do ônibus (GPS, WGS84)
DataHora2	Data e hora da antepenúltima coleta de dados do GPS do ônibus
Latitude2	Antepenúltima latitude conhecida do ônibus (GPS, WGS84)
Longitude2	Antepenúltima longitude conhecida do ônibus (GPS, WGS84)
DataHora3	Data e hora da coleta de dados do GPS do ônibus anterior a 2
Latitude3	Latitude conhecida do ônibus da coleta anterior a 2
Longitude3	Longitude conhecida do ônibus da coleta anterior a 2
DataHora4	Data e hora da coleta de dados do GPS do ônibus anterior a 3
Latitude4	Latitude conhecida do ônibus da coleta anterior a 3
Longitude4	Longitude conhecida do ônibus da coleta anterior a 3

Tabela 7 – Arquivo últimas 5 posições conhecidas de cada ônibus

Este arquivo é separado por tabulação (caractere "\t"), ordenado pelo identificador alfanumérico dos ônibus (campo denominado "Ordem") e não possui cabeçalho. A medida que uma nova posição é recebida, ou seja, uma latitude, longitude, data e hora distintas da última posição

conhecida, essa é armazenada como a última conhecida e a que anteriormente era a última passa a ser a penúltima, a penúltima passa a ser a antepenúltima e assim sucessivamente. São armazenadas sempre as últimas cinco posições distintas conhecidas de cada ônibus neste arquivo. O tamanho deste arquivo tende a evoluir lentamente ao longo do tempo, em função da entrada de novos ônibus em circulação. Ônibus que são tirados de circulação permanecem no arquivo com suas últimas posições conhecidas.

- **Histórico de posições dos ônibus ("{aaammdd}.zip"):** Arquivo em formato texto que salva cada nova posição conhecida em sua última linha. Sabe-se que uma posição é nova ou não a partir do arquivo anteriormente apresentado. Ao final de cada dia o arquivo é compactado e nomeado com a respectiva data e um novo arquivo em branco é criado.

Campo	Descrição
DataHora	Data e hora da coleta de dados do GPS do ônibus
Ordem	Identificação alfanumérica encontrada na lateral dos ônibus
Linha	Identificador da linha do ônibus
Latitude	Latitude do ônibus (GPS, WGS84)
Longitude	Longitude do ônibus (GPS, WGS84)
Velocidade	Velocidade do ônibus na hora da coleta

Tabela 8 – Arquivo histórico de posições dos ônibus

Este arquivo também é separado por tabulação (caractere "\t") e a ordenação é cronológica em função do recebimento dos dados. O mesmo não possui linha de cabeçalho, mas para cada novo conjunto de posições, ou seja, que são frutos de uma mesma leitura do conjunto de dados Posição dos Ônibus, é acrescenta uma linha de separação com o formato "# x dado(s) novo(s) inserido(s) em dd-MM-yyyy HH:mm:ss ", onde o "x" é substituído pelo total de linhas inseridas com novas posições e "dd-MM-yyyy HH:mm:ss" pelo momento em que ocorreu a leitura do conjunto de dados, sendo esse com precisão de segundos. Em função do volume de ônibus em circulação e da atualização de sua posição, ao longo do dia, a cada 1 minuto, esse arquivo cresce. A Tabela 8 descreve resumidamente as informações armazenadas nesse arquivo.

- **Linhas de ônibus (“lines.txt”):** Arquivo em formato texto que contêm em cada linha informações sobre cada linha de ônibus, conforme dados especificado na Tabela 9.

Campo	Descrição
Linha	Identificador da linha do ônibus
Descrição	Texto apresentado no letreiro dos ônibus desta linha
ShapesId	Identificador dos trajetos da linha separado por ponto e vírgula
TemShape	True se possui trajeto conhecido e False caso contrário
TemParadas	True se possui paradas conhecidas e False caso contrário

Tabela 9 – Arquivo linhas de ônibus

Este arquivo também é separado por tabulação (caractere “\t”) e ordenado pelo identificador da linha de ônibus (campo denominado “Linha”). O mesmo não possui linha de cabeçalho. A cada novo recebimento de informações sobre as linhas, as informações armazenadas são apagadas e as novas informações recebidas são armazenadas. O tamanho deste arquivo evolui de forma proporcional à quantidade de linhas de ônibus em operação, sendo as diferentes linhas de ônibus em operação identificada a partir do campo Linha do conjunto de dados Posição dos Ônibus.

- **Pontos de parada das linhas de ônibus (“stops.txt” e “stop_{linha}.txt”):** Arquivo em formato texto que contêm em cada linha informações sobre uma parada de ônibus, conforme dados especificados na Tabela 10.

Campo	Descrição
Linha	Identificador da linha do ônibus
Sequência	Número da parada que é um sequencial para cada linha
Latitude	Latitude do ponto de parada de ônibus (GPS, WGS84)
Longitude	Longitude do ponto de parada de de ônibus (GPS, WGS84)

Tabela 10 – Arquivo pontos de parada das linhas de ônibus

Este arquivo também é separado por tabulação (caractere “\t”) e ordenado pelo identificador da linha de ônibus (campo denominado “Linha”). Além disso, o mesmo não possui linha de cabeçalho. A cada novo recebimento de informações sobre as paradas, estas são

armazenadas e apagadas para o armazenamento de novas informações recebidas.

- **Trajetos da linha de ônibus ("shape_{linha}.txt"):** Arquivo em formato texto que contêm em cada linha posições geográficas referentes a um trajeto da linha de ônibus, conforme dados especificado na Tabela 11.

Campo	Descrição
Sequência	Número da coordenada em relação ao trajeto ao qual pertence
ShapeId	Identificador do trajeto ao qual essa coordenada pertence
Latitude	Latitude da coordenada (GPS, WGS84)
Longitude	Longitude da coordenada (GPS, WGS84)

Tabela 11 – Arquivo pontos de parada das linhas de ônibus

Este arquivo também é separado por tabulação (caractere "\t") e ordenado pelo identificador do trajeto (campo denominado "ShapeId"). Além disso, o mesmo não possui linha de cabeçalho. A cada novo recebimento de informações sobre os trajetos, as informações armazenadas são apagadas e as novas informações recebidas são armazenadas.

Todos esses arquivos são armazenados como Blobs, com o grau de redundância desejado. Os arquivos **shape_{linha}.txt**, **stops.txt, stop_{linha}.txt** e **lines.txt** ficam na pasta "csv" e os três primeiros são zipados diariamente para armazenamento do histórico de evolução das linhas no formato "aaaammdd.zip" dentro desta pasta. Os arquivos **{aaammdd}.zip** e **now.txt** ficam na pasta "gps". O arquivo **gtfs.zip** fica na pasta gtfs, sendo armazenado uma cópia diária com a data do dia no formato "aaammdd.zip" dentro desta pasta.

Possivelmente, para testes do processo de coleta, o acesso a estes arquivos pode ser realizado pela Internet, bastando a configuração da pasta como pública e o uso do endereço: *http://{conta}.blob.core.windows.net/{pasta}/{arquivo}*. Onde {conta} corresponde a criada no serviço *Microsoft Azure Storage Blob* (e.g. "plataforma"), {pasta} é a coleção criada neste serviço para armazenar estes arquivos (no caso dos dados de GPS seria "gps" e nos demais "csv") e o {arquivo} seria o arquivo supracitado (e.g. now.txt).

O procedimento de coleta, possivelmente, pode gerar dados de *log,* gravados em banco de dados, conforme a especificação descrita na Tabela 12.

Campo	Descrição
DataHora	*Timestamp* da gravação do registro com precisão de milissegundo
Tipo	Tipo do registro (W – Alerta; I – Informação; E – Erro)
Mensagem	Mensagem com até 250 caracteres
Instância	Identificador da instância e da versão do serviço

Tabela 12 – Campos da tabela de log

3.2 Coleta dos Dados

Com objetivo de coletar os conjuntos de dados disponíveis no Portal (vide Capítulo 3) foi desenvolvido um serviço na nuvem que executa de forma cíclica o procedimento abaixo detalhado, sendo o serviço detalhado no capítulo 4.

1. Recupera da instância onde o serviço está sendo executado ou, se necessário, do armazenamento compartilhado, as últimas cinco posições conhecidas de cada ônibus, ou seja, o arquivo "now.txt" apresentado na seção 3.1. Aloca em memória os dados deste arquivo utilizando uma lista.
2. Obtêm o conjunto de dados com a **Posição dos ônibus** pela Internet em suas 3 (três) versões.
3. Para cada linha do conjunto de dados verifica se a posição, data e hora são diferentes da última posição conhecida, que se encontra na lista em memória. Caso seja diferente, atualiza essa nova posição, descartando a posição mais antiga dentre as cinco anteriormente conhecidas para este ônibus. Caso o ônibus não esteja na lista, insere com a posição recebida. Além disso, para toda nova posição, armazena a mesma em um vetor.
4. Ao final da leitura de todas as posições recebidas, utilizando a lista, salva o novo arquivo "now.txt" localmente e no armazenamento compartilhado e, utilizando o vetor, apensa no arquivo "{aaaammdd}.txt" referente ao dia os dados obtidos, diretamente no armazenamento compartilhado, incluindo um separador referente à leitura em questão, conforme especificado na seção 3.1.
5. Caso exista no armazenamento compartilhado um arquivo "{aaaammdd}.txt" do dia anterior ao atual e nenhum procedimento diário em execução, abre uma nova *thread* que:

   a. Compacta o arquivo "{aaaammdd}.txt" do dia anterior gerando, assim, o arquivo "{aaaammdd}.zip" no armazenamento compartilhado.
   b. Apaga o arquivo do dia anterior, que não estava compactado, do armazenamento compartilhado.
6. Caso faça mais de uma hora que foram obtidos os conjuntos de dados de trajetos e paradas, abre uma nova *thread* que executa os passos a seguir. Teoricamente esses dados só são atualizados de um dia para o outro, mas como o esforço computacional é mí nimo, optamos pela periodicidade de hora em hora para atualizá-los.
   a. Obtêm os dados de linhas a partir do arquivo "lines.txt" do armazenamento compartilhado e carrega numa lista em memória.
   b. Obtêm todas as linhas diferentes em operação a partir do arquivo "now.txt" do armazenamento compartilhado e adiciona as linhas que não existiam na lista anteriormente obtida.
   c. Cria um novo arquivo "stops.txt" localmente.
   d. Para cada linha de ônibus da lista, obtêm o conjunto de dados **Pontos de parada das linhas do ônibus** da Internet e insere os pontos de parada, um a um, no final do arquivo local "stops.txt". Cria subconjuntos "stops_{linha}.txt". Envia estes arquivos para o armazenamento compartilhado, subscrevendo o anterior.
   e. Para cada linha de ônibus da lista, obtêm o conjunto de dados **Coordenadas dos trajetos das linhas de ônibus da Internet** e deleta o arquivo local "shape_{linha}.txt" referente a linha e cria um novo, inserindo as coordenadas de todos os trajetos desta linha. Envia este arquivo para o armazenamento compartilhado subscrevendo o anterior.
   f. Se ainda não salvou uma versão das linhas e dos paradas para o dia, salva em "shapes.zip" e "stops.zip", contendo os respectivos arquivos. Também obtêm e salva a versão do GTFS para o dia, assim como estes arquivos de forma compactada referente ao dia. Envia os arquivos para o armazenamento compartilhado.
7. O procedimento aguarda até completar 40 segundos de execução, caso ele não tenha levado este tempo para executar. Como cada dispositivo

de GPS envia dados de forma assíncrona, um dado novo com a posição do ônibus pode chegar a qualquer instante, mas um mesmo dispositivo só envia novamente um novo dado ao completar 1 minuto do último envio, portanto, qualquer valor inferior a esse tempo garante a captura de todos os dados transmitidos. Por outro lado, leituras em intervalos muito pequenos podem ocasionar bloqueio de acesso por parte da Prefeitura aos dados. Frente ao exposto, optamos por aguardar 40 segundos, o que na prática pode conferir um atraso máximo de 80 segundos na posição do ônibus, o que ocorre quando uma leitura é feita imediatamente após o envio de dados por um dispositivo de GPS. Esse atraso é irrelevante para análises históricas, mas pode representar um deslocamento na ordem de 500 metros para soluções em tempo real. Lembrando que a execução das *threads* abertas (item 5 e 7) ocorrerá em paralelo ao restante do procedimento. Retorna ao passo 1.

Cabe observar que o procedimento implementa um controle de *threads* baseado na sinalização do momento de início da execução, que é desmarcada após seu término. Além disso, o tempo máximo de execução de cada *thread* é controlado, permitindo assim, que ela seja reiniciada se não finalizada no tempo esperado, o que funciona como um sistema de tolerância a falha.

3.3 Visualização Básica dos Dados

Visando compreender um pouco melhor os dados disponíveis e validar a coleta destes, foram desenvolvidas algumas visualizações básicas, utilizando a API (*Application Programming Interface*) do Google Maps [81] para sobrepor estes dados em um mapa personalizado disponibilizado pelo Google. Além disso, foram desenvolvidos alguns filtros básicos para que diferentes combinações destes dados pudessem ser visualizadas. Para tanto, foram utilizadas as linguagens PHP5, HTML e Java Script, sendo essa uma página Web, que consume a citada API.

A imagem a seguir apresenta a tela de filtros desenvolvida que além de escolher os critérios da visualização, concentra os principais links para os arquivos brutos armazenados. Na sequencia são também apresentadas

visualizações obtidas a partir desta ferramenta, com diferentes critérios de filtro (Veja Figura 6).

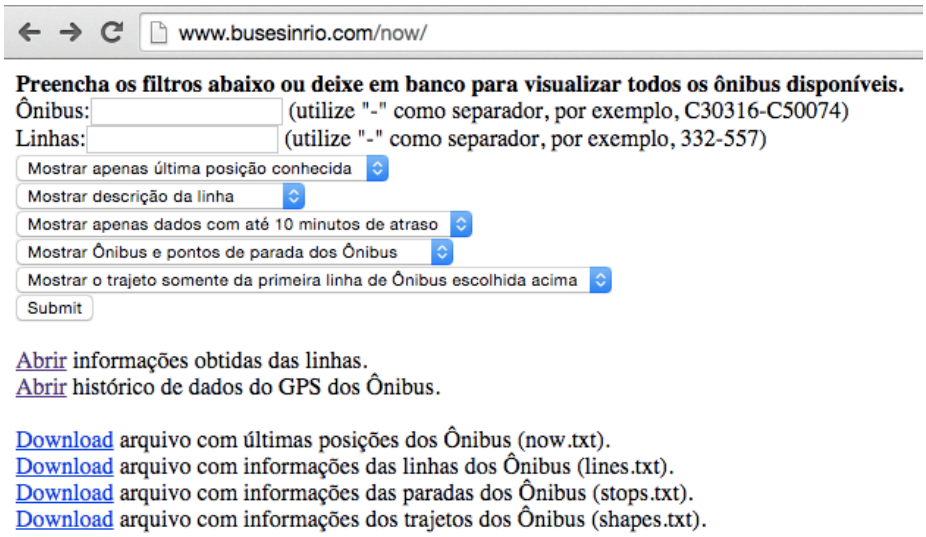


Figura 6 – Tela de Filtros Desenvolvida para Visualização Básica

A Figura 7 ilustra a visualização da última posição conhecida de todos os ônibus. Ao clicar em um ônibus são apresentados os dados coletados. Em verde estão os ônibus convencionais e em azul os do sistema BRT. Os em vermelho não possuem linha informada.

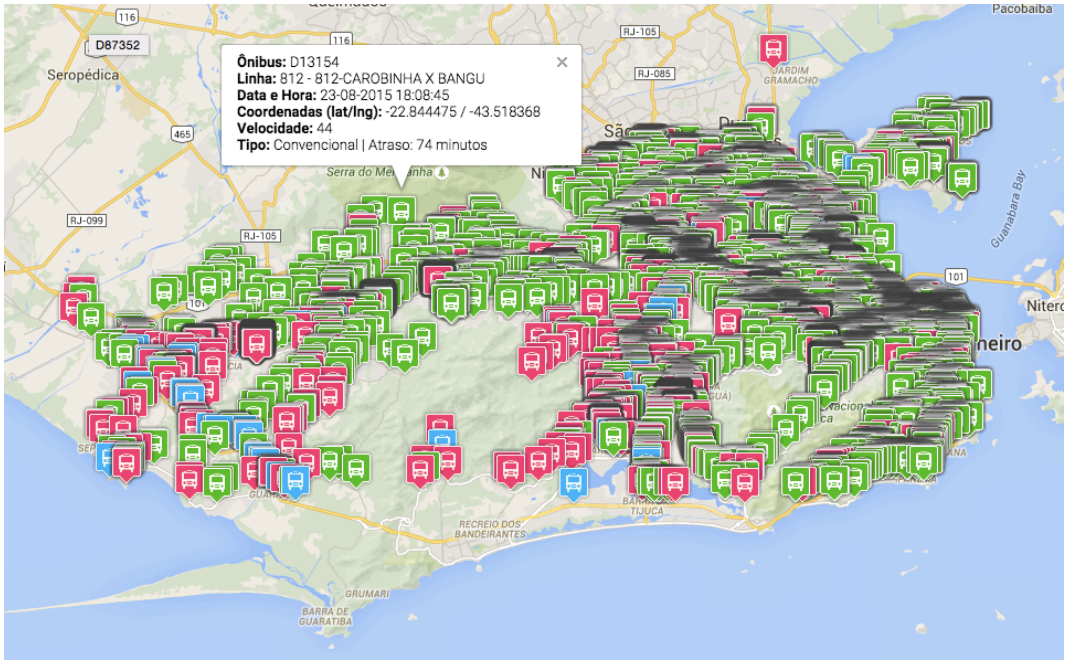


Figura 7 – Visualização da última posição conhecida

A Figura 8 ilustra a visualização da última posição conhecida de todos os ônibus com até 10 minutos de atraso em relação ao momento de geração da

visualização. É possível observar que no momento em que a imagem foi gravada o sistema BRT estava fora do ar, pois nenhum ônibus foi apresentado.

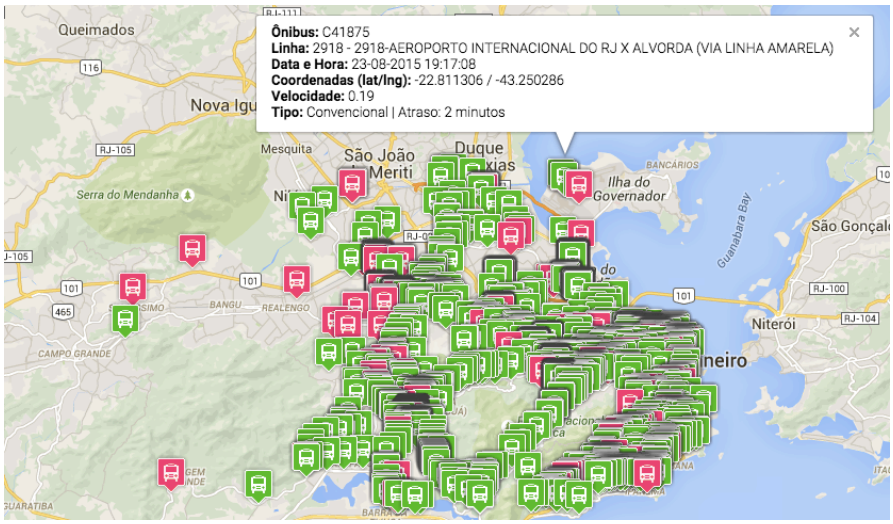


Figura 8 – Visualização da posição com até 10 minutos de atraso

A Figura 9 ilustra a visualização da última posição conhecida dos ônibus da linha 125 com até 10 minutos de atraso. Também apresenta os trajetos desta linha, que como é fácil perceber, são dois, um em vermelho e outro em azul, provavelmente representando o trajeto de ida e volta da linha.

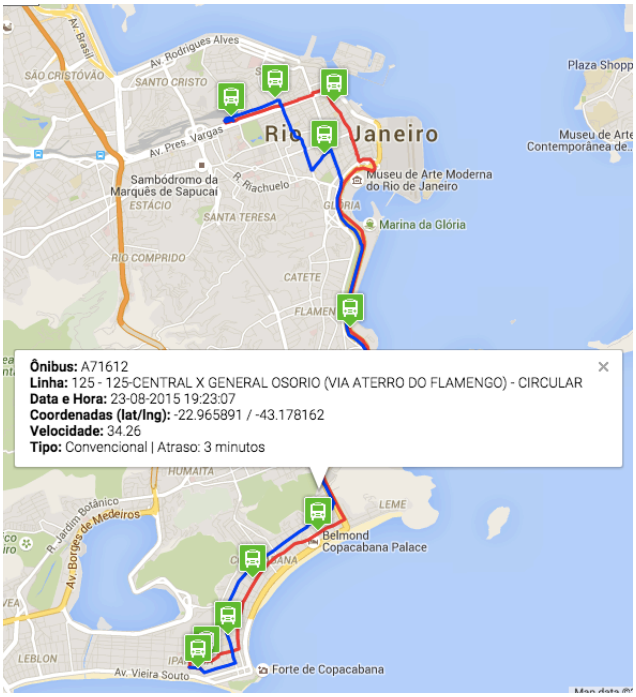


Figura 9 – Visualização da linha 125

As figuras que se seguem variam apenas o número da linha em relação a anterior, sendo agora de interesse a linha 415 (veja Figura 10) e em seguida a 390, com (veja Figura 12), e sem paradas (veja Figura 11).

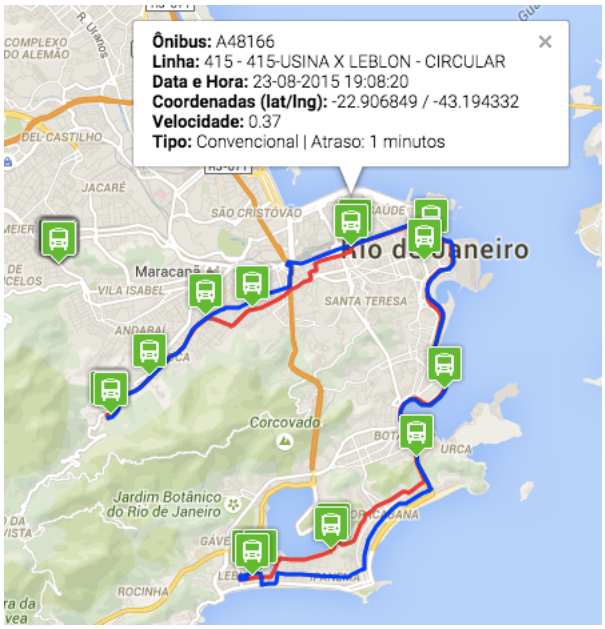


Figura 10 – Visualização da linha 415

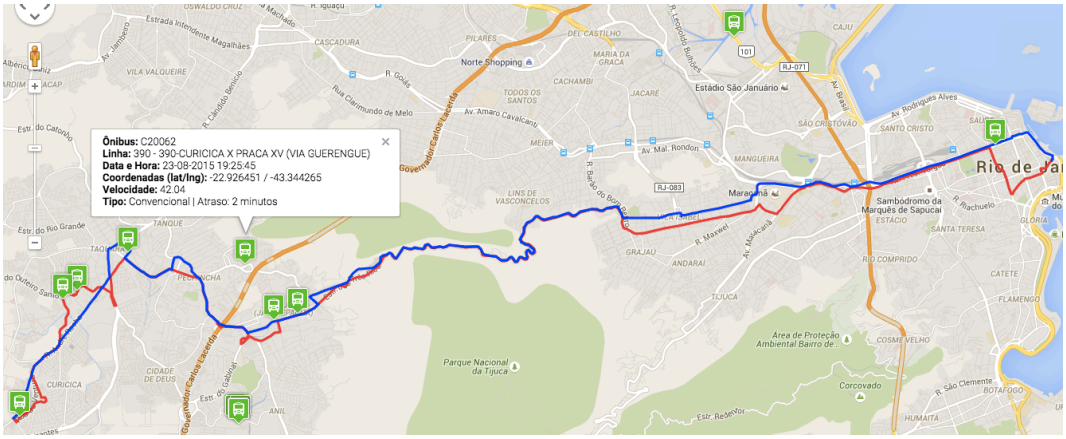


Figura 11 – Visualização da linha 390

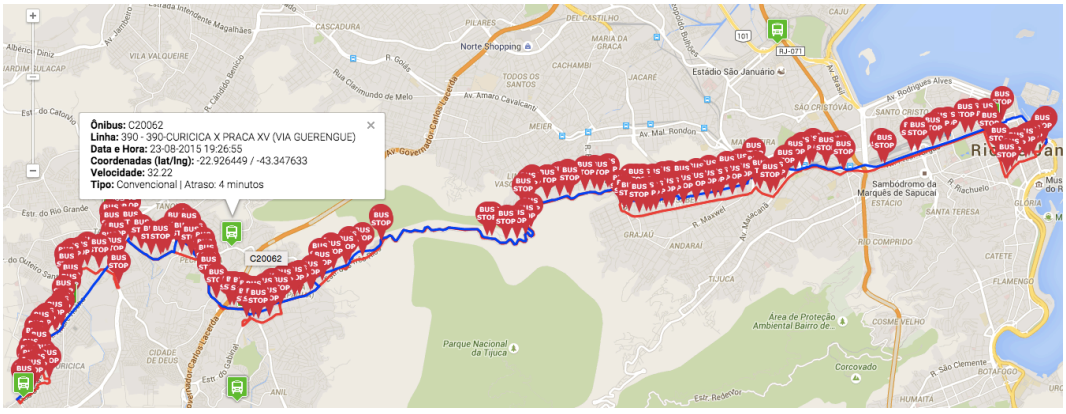


Figura 12 – Visualização da linha 390 com pontos de parada

Cabe registrar que os dados de parada aparentaram inconsistência em todas as linhas, dada a pouca distância entre eles, não se mostrando úteis.

Por último, a Figura 13 apresenta as 5 últimas posições de um ônibus, em particular, o A48081, da linha 415, cujo trajeto também é apresentado.

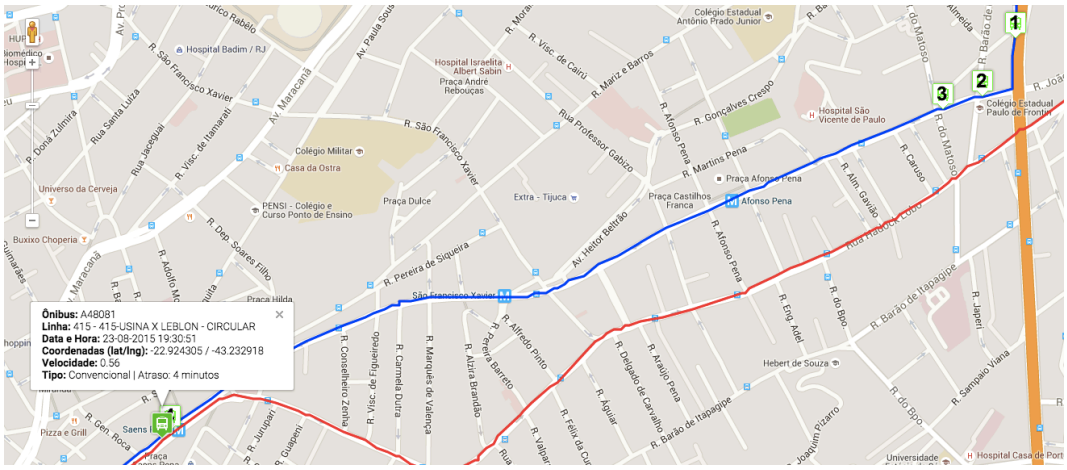


Figura 13 – Visualização das 5 últimas posições

4
A Plataforma

4.1
Objetivo

A Plataforma BusesInRio desenvolvida tem como objetivo facilitar a extração de conhecimentos a partir de dados de localização georeferenciada de mobilidade urbana, em particular, de dados do serviço de transporte público via ônibus municipal da cidade do Rio de Janeiro.

4.2
Computação na Nuvem

Visando conferir escalabilidade, elasticidade e estabilidade para Plataforma BusesInRio desenvolvida, assim como, não se preocupar com sua infraestrutura (e.g. *hardware*, sistema operacional, *softwares* básicos, etc.), optou-se pela utilização de um ambiente de computação na nuvem.

A ideia essencial da computação na nuvem é permitir o consumo de recursos computacionais, e.g., armazenamento, processamento, banda de entrada e saída de dados, sejam realizados através de serviços, com pagamento baseado na utilização dos serviço e grande elasticidade, i.e. capacidade de provisionar e desprovisionar rapidamente grandes quantidades de recursos em tempo de execução.

O Windows Azure [58] é a proposta da Microsoft para serviços de computação na nuvem. Os principais motivos que conduziram a essa escolha são resumidos a seguir.

1) Abstração do sistema operacional, pois, a princípio, não há necessidade de instalar nada nos servidores, o que não ocorre com outros fornecedores;

2) A IDE (*Integrated Development Environment*) exigida para desenvolvimento de aplicativos é gratuita (*Miscrosoft Visual Studio Community 2015*) e fornece uma série de facilidades para o desenvolvedor, tais como autocompletar, documentação dos métodos, dentre outras;

3) Fácil instalação dos requisitos e kits para desenvolvimento e testes em máquina local, reduzindo o tempo e custo de desenvolvimento;

4) Disponibilidade de uma grande variedade de artigos, exemplos, livros e materiais de formação;

5) A escolha da linguagem de programação C#.Net aproveita o conhecimento da sintaxe do C/C++, que muitos desenvolvedores dominam;

6) Interoperabilidade com outras tecnologias e linguagens de programação, através do WCF (*Windows Communication Foundation*);

8) Oferece a possibilidade de armazenar arquivos de forma compartilhada, com redundância e estabilidade, através do serviço *Azure Blob Store*;

9) Dispõe do sistema de gerenciamento de banco de dados, relacional, *Azure SQL*, que suporta dados espaciais e linguagem padrão de consulta;

10) Possui um banco de dados orientado a documentos, *Azure DocumentsBD*, que suporta dados semiestruturados;

11) Detêm um leque abrangente de configurações, permitindo a escolha dos recursos computacionais sob medida para cada situação;

12) Por último, mas não menos importante, oferece suporte a pesquisas selecionadas (i.e. recursos computacionais gratuitos).

4.3 Estratégia de Armazenamento e Análise dos Dados

Um dos maiores dilemas enfrentados no desenvolvimento desta Plataforma BusesInRio foi a definição da melhor estratégia para armazenamento, processamento e análise dos dados, uma vez que essa definição influencia significativamente o custo financeiro e computacional, podendo comprometer a sustentabilidade da solução e, especialmente, sua performance.

Abaixo elencamos os principais requisitos não funcionais da Plataforma BusesInRio, relevantes ao contexto em tela:

1. A Plataforma deve persistir todos os dados obtidos da Prefeitura.
2. A Plataforma deve informar o mais próximo possível do tempo real.
3. A Plataforma deve minimizar o tempo de resposta a requisições.
4. A Plataforma deve priorizar a otimização de custos.
5. A Plataforma deve evitar riscos de perda de dados e indisponibilidade.
6. A Plataforma deve ser escalável.

De forma a organizar o raciocínio, dividimos as principais opções disponíveis de armazenamento em grupos, conforme apresentado a seguir.

- **RAM**

  Os dados armazenados na memoria principal (*Randon Access Memory*) da instância, sem sombra de dúvida, possuem o melhor desempenho computacional, no entanto, este é um espaço mais custoso, limitado e sem persistência (i.e. os dados são perdidos ao reiniciar a instância). Portanto, deve ser usada quando o custo se justificar e a persistência não se apresentar como necessária. Tanto as máquinas virtuais (VM), quanto os serviços de nuvem (CS), disponibilizam este recurso computacional em diferentes medidas.

- **BD SQL**

  Os sistemas de gerenciamento de banco de dados relacionais com arquitetura integrada com extensões espaciais permite o controle e manipulação de objetos espaciais, com gerência de transações, controle de integridade, concorrência e linguagens próprias de consulta.

  O PostGIS [59] que estende o PostgreSQL é o software livre mais popular deste grupo. Suas características são exploradas em profundidade em [60].

  O PostGis não é um serviço oferecido pelo fornecedor em questão, no entanto, pode ser instalado em uma máquina virtual, consumindo seu disco local. Dessa forma, redundância, segurança, performance e manutenção ficam por conta do desenvolvedor.

  Outra opção deste grupo, é o serviço do fornecedor, o Azure SQL, uma vez que contempla garantias de performance, escalabilidade e redundância, também suportando operações espaciais.

  Um ponto que merece registro, é que em ambos os cenários é possível particionar tabelas, para melhorar a performance. Além disso, seus principais diferenciais são o suporte aos padrões do *Open Geospatial Consortium (OGC)*, o que facilita o intercâmbio de informações geográficas, e a possibilidade de definir tipos de dados espaciais,

equipados com operadores específicos (operadores topológicos e métricos).

- **BD NoSQL**

  Os bancos de dados orientados a documentos são bastante diferentes dos tradicionais bancos de dados relacionais. Em vez de armazenar dados em estruturas rígidas, como tabelas, eles os armazenam em documentos vagamente definidos [61].

  O MongoDB [62] é o mais populares deste grupo, destacando-se pelo suporte a operações espaciais. Segundo [63], este possui desempenho superior ao PostGIS, mas com um conjunto mais limita de operações espaciais disponíveis.

  Este serviço também não é oferecido pelo fornecedor em questão, de igual forma, pode ser instalado em uma máquina virtual. Em particular, existem imagens prontas para sua configuração, tornado fácil configurar a tecnologia, mas, novamente, questões como segurança, performance, redundância e escalabilidade ficam sobre a responsabilidade do desenvolvedor.

  De forma análoga ao grupo anterior, o fornecedor oferece sua tecnologia similar como um serviço, o DocumentBD, com os benefícios claros de disponibilidade, escalabilidade e redundância. Esse, como no caso anterior, obedece um modelo de pagamento por faixas de uso mensal.

  Cabe comentar que este grupo foi concebido para funcionamento em múltiplas instancias e para processamento de grandes volumes de dados semiestruturados (i.e. a estrutura pode variar entre documentos ou ser modificada ao longo do tempo).

- **BLOB**

  O fornecedor disponibiliza de forma nativa um serviço de armazenamento de blobs (“Binary Large OBject“), que nada mais são do que qualquer objeto binário, como áudios, vídeos e documentos, incluindo-se arquivos no formato TXT e ZIP.

  Capacidade e escalabilidade massiva, com o pagamento apenas pelo uso do espaço e transações realizadas, contando ainda com alta disponibilidade, redundância e segurança/criptografia, são algumas das

características do serviço, que intitula-se ideal para armazenar Big Data para análise.

A grande questão neste serviço é que os dados precisam ser transferidos para uma instância para serem manipulados, uma vez que nenhuma operação é suportada no serviço, muito menos operações espaciais.

Estes grupos foram primeiramente estudados com o enfoque de persistir todos os dados recebidos da Prefeitura, e em especial, os dos dispositivos de GPS dos ônibus, que são preponderantes em termos de volume, apresentando, ainda, crescimento significativo diária. Como referência, somente o dia 02/09/2016 acresceu 6.263.744 de registros distintos enviados pelos equipamentos de GPS.

Na captura de dados descrita no capítulo 3, os dados de GPS foram armazenados no Azure Blob Store. Como referência, esses dados de 02/09/2016 totalizam 86,18 MB se compactados (formato ZIP), aumentando para 331 MB se armazenados em texto corrido não compactado (formato TXT). Quando inseridos no PostGIS, Azure SQL, MongoDB ou DocumentBD, o espaço requerido é muitas vezes maior do que para os dados em texto corrido, conforme observado em todas as manipulações dos dados diários nestas tecnologias (e.g. o dia 02/09/2016 ocupou 1.04 GB no DocumentDB e apenas 86,18 MB em arquivo compactado, enquanto o dia 02/06/2015 ocupou 1,51 GB no PostGis e apenas 85,88 MB em arquivo compactado). Via de regra, para simplificar, podemos manter em mente a relação de 1 para 10 do arquivo compactado para o espaço requerido nestes banco de dados, ou seja, se projetarmos um espaço de 100 GB para o grupo BLOB, teríamos que projetar na ordem de 1 TB nos grupos BD SQL e BD NoSQL. feitos diversas Lembrando que em agosto de 2016 apenas os arquivos compactados totalizavam mais de 36 GB, o que torna 100 GB uma projeção adequada para o horizonte de um ano, dado que outras informações serão armazenadas.

A Tabela 13 lista as tecnologias supracitadas, adicionado referências de custo obtidas em 03/09/2016 do site do fornecedor [58] para uma mesma região.

Grupo	Tecnologia	Capacidade (GB) x Mensal (R$/mês)
BD SQL	Azure SQL	2 GB por R$18,69/mês
RAM	Azure Cloud Service	0,75 GB por R$55,80/mês
Blob	Azure Blob Storage	1.000 GB por R$90,07/mês (c/ transações e pago pelo uso)
BD SQL	PostGIS na VM	1.000 GB por R$5.169,87/mês
BD NoSQL	MongoDB na VM	1.000 GB por R$5.169,87/mês
BD NoSQL	Azure DocumentosDB	1.000 GB por R$5.401,50/mês (c/ transações e pago pelo uso)
BD SQL	Azure SQL	1.024 GB por R$26.253,90/mês
RAM	Azure VM	448 GB por R$26.923,87/mês

Tabela 13 – Tecnologias para armazenamento e processamento

Fica evidente que a opção de menor custo para armazenamento de grandes volumes de dados é o Blob, mesmo se considerarmos que o espaço requerido para armazenar os dados é igual entre os grupos. No entanto, conforme exposto anteriormente, o espaço requerido nos grupos BD SQL e BD NoSQL é maior para os mesmos dados diários compactados armazenados em BLOB. De forma complementar, ao utilizarmos o Azure Blob, o fornecedor garante redundância de dados e do serviço, reduzindo riscos, assim como, assegurando escalabilidade, ou melhor ainda, elasticidade (i.e. possibilidade de consumir mais ou menos serviço ao longo do tempo). Resta agora avaliarmos a performance (tempo de resposta / tempo real) deste armazenamento.

Ocorre que, conforme registrado na descrição do grupo RAM, a memória principal é sem dúvida a mais rápida, portanto, é inócuo comparar sua performance frente ao Azure Blob. No entanto, dado seu custo extremamente elevado, é preciso refletirmos em que situações essa performance é requerida e o custo associado se justifica.

Nesse sentido, se pretendemos informar em tempo real, ou pelo menos, em tempo "quase" real, dado que algum atraso é intrínseco ao próprio dado recebido, justifica-se buscarmos essa velocidade, desde que não se perca de vista a preocupação com custo. Isto posto, temos que atentar que somente as informações mais recentes são relevantes nesse contexto, pois o passado, por definição, não trata do que ocorre naquele instante na realidade. Dessa forma, o uso da memoria principal pode ser conveniente, mantendo-se o cuidado com o volume armazenado na mesma para processamento, buscando assim, uma boa relação de custo-benefício. Em particular, podemos implementar um serviço na nuvem, a um

custo mínimo, especificamente com essa finalidade de armazenar e manipular em memória o subconjunto de últimas informações recebidas de cada ônibus (ex. 10 últimas posições) e linha (descritização atual das suas trajetórias e demais informações, como tarifa e nome), descartando as informações na medida que se distanciem do momento corrente, assim como, relacionando-as com conhecimentos previamente extraídos do passado. Nessa estratégia, como os dados utilizados estarão, em sua maioria, em memória principal, e nela serão processados, será alcançado, como consequência, um excelente tempo de resposta. Em função desta estratégia, que precisa apenas de um subconjunto enxuto de dados, um serviço na nuvem com pouca memória já apresentaria bons resultados. Vamos aprofundar a definição deste serviço na seção 4.4.

Do ponto de vista do processamento/análise dos dados históricos, tempo real por definição não é uma preocupação. Porém, os serviços na nuvem também são interessantes para assegurar tempo de resposta adequado, no entanto, não podem armazenar a totalidade dos dados históricos, obviamente, em função do custo, sendo aceitável, no máximo, realizar o *cache* de uma ou outra informação ou levar um subconjunto para memória em busca de desempenho de processamento. Combinando esses serviços com o armazenamento em BLOB, poderíamos extrair conhecimento destes dados, uma vez que qualquer operação, espacial ou não, pode ser implementada nos serviços, com ampla liberdade. Em outras palavras, os arquivos diários podem ser copiados do armazenamento em BLOB para a instância do serviço na nuvem que o processará, sendo o mesmo descartado posteriormente, e o conhecimento extraído salvo em armazenamento adequado, a ser discutido oportunamente. Lembrando que essa transferência de arquivos (BLOB) ocorre em rede local de alta velocidade, desde que o serviço e armazenamento sejam configurados na mesma região. Como referência, um arquivo compactado, como o dia 02/09/2016 supracitado, leva menos de 2 segundos para ser transferido do BLOB para a instância. Além disso, fica evidente que o processamento do arquivo se dará de forma sequencial, linha a linha, portanto, quanto maior o conjunto de conhecimentos que puderem ser extraídos em uma única passagem, menor será o ônus desta transferência e leitura.

Em contrapartida, tendo em vista que tempo de resposta é a questão em tela, faz sentido avaliarmos o uso dos grupos BD SQL e BD NoSQL. Conforme estudo

supracitado, podemos considerar que o BD NoSQL seria uma alternativa de maior desempenho. Em particular, aprofundamos nossos experimentos no DocumentBD, uma vez que é a solução nativa deste grupo, ou seja, não perderíamos a escalabilidade e ao mesmo tempo não imputaríamos riscos adicionais em relação ao BLOB.

Um único registro do dispositivo de GPS de ônibus enviado pela Prefeitura pode ser entendido como um documento, como exemplificado na Figura 14.

```
{
  "id": "B32501_20160902_000000",
  "DataHora": "2016-09-02T00:00:00",
  "Ordem": "B32501",
  "Linha": "SV779",
  "Location": {
    "type": "Point",
    "coordinates": [
      -22.87353,
      -43.34323
    ]
  },
  "Velocidade": "45"
}
```

Figura 14 – Um registro do dispositivo de GPS.

Imaginando que a coleção intitulada “gps” agrega estes documentos, poderíamos efetuar consultas sobre esses dados. Na verdade, um ponto interessante, é que essa tecnologia suporta consultas estilo SQL (nem todos os comandos são suportados, por exemplo, não suporta agregação). Além disso, também suporta o *Open Geospatial Consortium* (OGC) para consultas espaciais. Alguns exemplos de consulta são apresentados a seguir.

- SELECT * FROM gps WHERE gps.DataHora >= "2016-09-02T03:00:00" AND gps.DataHora < "2016-09-02T04:00:00"
- SELECT TOP 1 gps.DataHora FROM gps WHERE gps.Ordem = "B25512" ORDER BY gps.DataHora DESC
- SELECT gps.Ordem FROM gps WHERE gps.Linha = " SV232"
- SELECT * FROM gps WHERE gps.Linha = "950" AND ST_DISTANCE(gps.Location,{"type": "Point","coordinates": [-22.81, -43.30]}) < 1000 -- (1km)

O tempo de resposta destas consultas é extremamente rápido (< 1 segundo), principalmente se comparada a estratégia anterior com consultas isoladas em

múltiplas datas, dado que envolve transferência de arquivos e leitura sequencial, podendo ainda ser reduzindo acrescendo índices e outros recursos disponíveis. No entanto, precisamos nos questionar se essas são a natureza de consultas que precisaremos para extrair conhecimento dos dados e se o tempo de resposta as consultas é o que determina o tempo de resposta ao usuário da Plataforma.

Nesse momento, torna-se extremamente relevante extrapolarmos a análise isolada dos grupos e voltarmos nossa visão para os processamentos do histórico que serão endereçados neste trabalho. Estamos, portanto, frente a uma conceituação relevante para a sequencia deste trabalho, a Plataforma BusesInRio proposta não se destina a realizar consultas em um ambiente de Big Data, mas sim, a realizar Data Science, ou seja, extrair conhecimento a partir destes dados (analisar o Big Data). Sendo assim, por exemplo, consultas interessadas na situação particular de um ônibus de forma isolada são pouco relevantes em comparação a consultas sobre a evolução das trajetórias destes ônibus ao longo do tempo (durante uma hora, um dia ou um período qualquer). Cada novo dia de registro de dados somasse aos demais na visão histórica. Evidentemente, a imensa maioria de algoritmos a serem implementados apresentarão como característica o processamento cronológico e incremental dos dados, ou seja, processamento do mais antigo para o mais recente, incluindo a cada dia os dados nele recebidos. Reforçamos que não está sendo tratada a discussão da visão do presente, do tempo real, cuja estratégia de processamento e armazenamento foi definida anteriormente, no grupo RAM.

Naturalmente, os procedimentos a serem propostos para extrair conhecimento a partir dos dados históricos ocorrem em duas situações, quando um procedimento novo é definido e toda cronologia de dados precisa passar pelo procedimento ou, quando este procedimento está estabilizado, ou seja, já processou todo o passado, sendo necessário aplicar o procedimento apenas a cada dia, para os dados novos capturados do mesmo.

Imaginando a primeira situação, ao trazer os dados para a instância, no caso do BLOB, estaríamos assegurando igual performance para todas as instâncias, permitindo assim, um processamento paralelo eficiente. No DocumentBD, o número de transações por segundo acessando esta mesma coleção centralizada, passa a ser um limitador da performance desta paralelização. O mesmo vale para

segunda situação, mas nesse caso o volume de dados é bem menor, mas precisamos considerar o tempo de inserção dos dados no armazenamento, que também é relevante, no caso do banco de dados orientado a documentos. Por exemplo, em um dia com 5 milhões de registros, o que é muito comum, a inclusão levaria 27,78 horas, obtidos pela seguinte conta: 5.000.000 x 10kb por documento x 2 procedimentos (validar se pode inserir e inserir) / 1.000 (unidades de requerimento por segundo suportadas) / 360 (conversão para horas), teríamos / 3.600. Na prática, a inserção do dia 02/09/2016, sem qualquer índice, levou mais de 2 dias. Portanto, dado o menor custo e alta capacidade de paralelização, faz todo o sentido o uso do Azure Blob combinado com serviços na nuvem para processar os dados históricos para extração de conhecimentos.

Corroborando, o tempo de resposta ao usuário final não depende, a priori, da velocidade de extração dos conhecimentos, mas sim da velocidade de consulta dos conhecimentos já extraídos, imaginando que idealmente essa extração ocorrerá previamente, para assegurar um tempo de resposta adequado.

Diante destas análises, optamos por permanecer armazenando os dados no Azure Blob, levando-os a uma instância para processamento, sempre que necessário. Essa decisão garante custos bem menores e liberdade para implantação de qualquer algoritmo nas instâncias, inclusive algoritmos que assegurem o processamento em paralelo quando necessário. Se identificada a necessidade, o SQL Azure poderá ser utilizado para manipular os conhecimentos já extraídos e o Azure DocumentDB para consultas visando obter informações que não possam ser previamente processadas.

Antecipando a apresentação dos algoritmos de extração de conhecimento, a ser realizada no capítulo 5, o tempo observado para aplicar todos estes algoritmos para um novo dia é inferior a cinco minutos, podendo ainda ser o processamento paralelizado para manter este tempo com um número maior de conhecimentos objetivados, ou seja, a meia noite e cinco minutos todos os conhecimentos adquiridos a partir do histórica estarão considerando até a data anterior. Com o BD NoSQL e BD SQL, o tempo de inserir os dados já não permitiria tal tempo de resposta. Em resumo, o tempo de resposta do ponto de vista do usuário utilizando-se o BLOB é praticamente "imediato" (apenas o tempo de consulta da informação já processada e salva e sua transmissão a solução cliente da

Plataforma), assegurando ainda o endereçamento dos demais requisitos supracitados.

A Tabela 14 resume as percepções supracitadas.

Tecnologia	Resumo
Azure Cloud Service	Serão utilizadas instâncias pequenas, que tem um custo extremamente acessível (vide tabela anterior). Processamento em paralelo pode ser utilizado para reduzir tempo de extração de conhecimento. A memória principal pode ser utilizada para informações mais próximas possíveis do tempo real nos casos aplicáveis. Pode implementar estratégias de *cache*.
Azure Blob Storage	Armazenará os dados históricos. A priori, também armazenará os conhecimentos extraídos. Essa decisão tem como racional o custo muito mais acessível e a possibilidade de processamento prévio em paralelo.
Azure SQL	Apresenta-se como uma boa alternativa para armazenar os conhecimentos já extraídos, oferecendo um bom tempo de resposta em caso de crescimento do numero de usuários da Plataforma. Nesse momento, se optou por não utilizarmos, em função do uso apenas experimental da Plataforma, no entanto, introduzir o mesmo é extremamente trivial.
Azure DocumentosDB	Os algoritmos a serem implementados neste trabalho, assim como, a possibilidade de processamento prévio em paralelo, não justificaram o uso desta tecnologia. No entanto, em situações particulares de outros pesquisadores (e.g. consulta envolvendo vários dias para uma resposta que não pode ser processada previamente), seria fácil migrar parte dos dados históricos para este serviço e efetuar consultas sobre estes dados.

Tabela 14 – Resumo das percepções e decisões

4.4 Arquitetura

A Plataforma BusesInRio encontra-se estruturada como um conjunto de serviços na nuvem com papeis específicos, que atuam de forma coordenada para alcançar bons resultados em termos de performance, disponibilidade, eficácia e eficiência. A Figura 15 ilustra a arquitetura da Plataforma BusesInRio e na sequencia todos os seus componentes são detalhados.

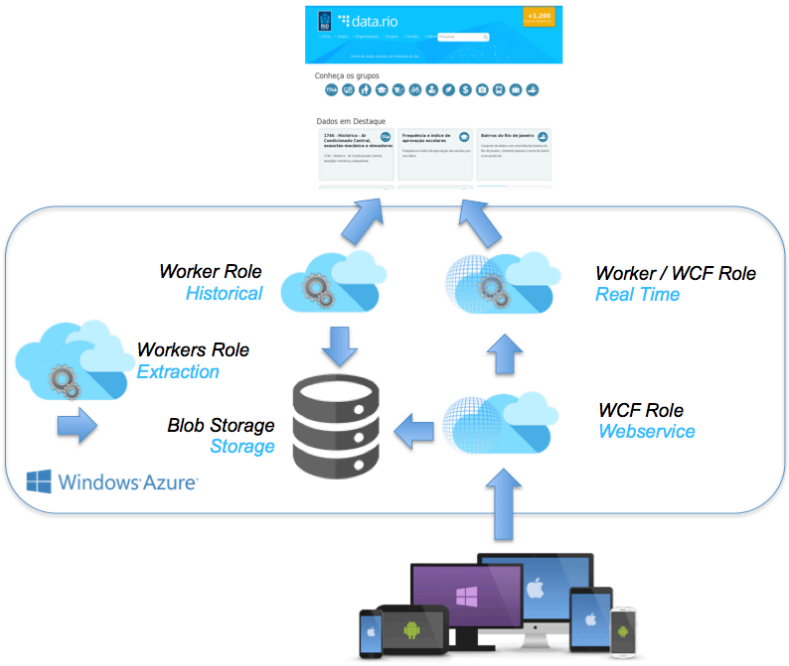


Figura 15 – Arquitetura da Plataforma na Nuvem

Conforme detalhado na seção 2.6, o conjunto de dados (i.e. Tabela 1 – Posição dos ônibus, Tabela 2 – Pontos de parada das linhas do ônibus, Tabela 3 – Coordenadas dos trajetos das linhas de ônibus e Tabela 4 – Arquivos do GTFS) é disponibilizado pelo Portal de Dados Abertos da Prefeitura do Rio de Janeiro, que é representado no topo da Figura 15 pela imagem de uma tela deste Portal. Esses conjuntos de dados são coletadas e armazenadas pelo serviço denominado ***Historical***. O algoritmo executado por este serviço foi detalhada na seção 3.2. Seu objetivo é assegurar que todos os dados disponibilizados sejam capturados e armazenados no serviço ***Storage***, formando assim um conjunto de dados histórico, que inclui os dados apresentados nas seções 2.6 e 3.1. Naturalmente, este serviço possui redundância e escalabilidade, reduzindo assim, o risco de qualquer perda de dados histórico.

Nenhum acesso externo é permitido a estes serviços da Plataforma BusesInRio, porém, cabe resgatar que para as visualizações básicas apresentadas na seção 3.3, foi liberado acesso direto aos arquivos do serviço *Storage*, uma vez que o objetivo na época era compreender os dados, mas na implantação definitiva da Plataforma os acessos foram centralizados, como será explicado adiante, para garantir a segurança, escalabilidade e controle da solução.

No outro lado da arquitetura, temos o serviço ***Real Time***, que tem como missão também coletar estes mesmos dados, mas esses não são salvos para

formação de um histórico, pelo contrário, são manipulados, estruturados e mantidos em memória principal como uma visão da situação atual das linhas e ônibus, ou seja, do que está acontecendo em tempo "quase" real. Essa questão foi discutida anteriormente.

Os serviços *Historical* e *Real Time* asseguram que não ocorrerão acessos desnecessários ao Portal. Estes não demandam escalabilidade, uma vez que apenas uma instância de cada serviço é suficiente para manter uma visão atualizada dos dados, tornando extremamente econômica sua manutenção. A demanda por elasticidade concentra-se na disponibilização destas informações, o que será endereçado no serviço *Webservice*.

Os serviços classificados como ***Extraction*** tem o objetivo de extrair conhecimento a partir dos dados históricos. Esse serviço pode ser implementado com várias funções (e.g. identificação de trajetórias) e se desejado com processamento em paralelo (e.g. cada instância processa um segmento). As implementações destes serviços são discutidas em detalhes no próximo capítulo.

Por sua vez, o serviço ***Webservice*** tem a missão de orquestrar o consumo dos dados históricos salvos em *Storage* e das informações atuais disponibilizadas pelo serviço *Real Time*, podendo o próprio ser escalado em função do número de usuários que o consomem, assim como, adotar estratégias de *cache*. Este serviço é também responsável pelo cruzamento das visões históricas e atuais. Este é o único canal de comunicação da Plataforma BusesInRio com o ambiente externo, sendo este o mesmo um serviço no padrão WCF [85] que implementa SOA (*service oriented architecture*), portanto, qualquer aplicação pode consumir este serviço.

4.5
Comunicação com a Plataforma

A Plataforma BusesInRio oferece através do serviço *Webservice* um conjunto de métodos para acesso aos conhecimentos extraídos a partir dos dados, são eles:

- `int HowManyBusesOperating()`
  Retorna o número de ônibus que estão operando naquele instante.

- `int HowManyRoutesOperating()`
  Retorna o número de linhas de ônibus que possuem pelo menos um ônibus operando naquele instante.

- `string[] GetRoutesStatus(int status)`
  Retorna o identificador (“linha”), nome, quantidade de ônibus operando naquele instante, quantidade de trajetos conhecidos e tarifa das linhas de ônibus. Os dados são separados por ponto e vírgula, sendo a informação de cada linha em uma posição do vetor retornado. Se o parâmetro “status” é enviado com 1, retorna apenas as linhas com pelo menos um ônibus operando naquele instante, se com 0, apenas as sem e, se com 2, todas, independente de possuir ou não ônibus operando naquele instante.

- `string[] GetBusesFromRoute(string linha)`
  Retorna o identificador (chamado de ordem) dos ônibus de uma linha de ônibus cujo identificador é enviado como parâmetro “linha”, sendo uma ordem em cada posição do vetor retornado.

- `string[] GetBuses(string[] ordem)`
  Retorna a latitude e longitude com a última posição conhecida dos ônibus cuja ordem são enviadas como parâmetro “ordem”, sendo a informação de cada ônibus em cada posição do vetor retornado e a latitude e longitude separadas por ponto e vírgula.

- `string[] GetAllBuses(int status)`
  Retorna a ordem de todos os ônibus. Se o parâmetro “status” é enviado com 1, retorna apenas ônibus operando naquele instante, se com 0, apenas os não operando e, se com 2, todos, independente se estão ou não operando naquele instante.

- `string[] GetBusDetails(string ordem)`
  Retorna a latitude e longitude das últimas 10 posições conhecidas do ônibus cuja ordem é enviada como parâmetro “ordem”, sendo uma posição em cada linha do vetor retornado e a latitude e longitude separadas por ponto e vírgula.

- `string[] GetBusDetailsWithProjection(string ordem)`
  Retorna a mesma estrutura do método anterior, no entanto, cada coordenada é projetada na trajetória da linha, caso o ônibus possua uma.

- `string[] GetShapesId(string linha)`
  Retorna o identificador de todos os trajetos da linha (“shape_id”) cujo identificador é enviado como parâmetro “linha”, estando cada identificador de trajeto em uma posição do vetor retornado.

- `string[] GetShape(string shape_id)`
  Retorna a latitude e longitude de cada ponto que distretiza a trajetória cujo identificador é enviado como parâmetro “shape_id”, sendo um ponto em cada linha do vetor retornado e a latitude e longitude separadas por ponto e vírgula.

- `string GetShapeLength(int id)`
  Retorna o comprimento em metros da trajetória da linha cujo identificador foi enviado no parâmetro “id”.

- `string[] GetStops(string shape_id)`
  Retorna a latitude e longitude de cada ponto de parada cujo identificador da trajetória da linha é enviado como parâmetro “shape_id”, sendo um ponto em cada linha do vetor retornado e a latitude e longitude separadas por ponto e vírgula.

- `string[] GetBusCloser(string latitude, string longitude, double distance)`
  Retorna a ordem dos ônibus próximos a uma coordenada, que seja enviada como parâmetros “latitude” e “longitude”, sendo um ônibus em cada linha do vetor retornado. O parâmetro “distance” define qual a distância máxima na qual um ônibus deve ser considerado próximo a coordenada enviada.

- `string[] GetSegments()`
  Retorna o identificador e nome de cada segmento monitorado, sendo um segmento em cada linha do vetor retornado e as informações em cada linha separadas por ponto e vírgula.

- `string GetSegmentRegion(int id)`
  Retorna a região do segmento cujo identificador foi enviado como parâmetro em “id”. A região é formada pela latitude mínima, latitude máxima, longitude mínima e longitude máxima, separadas por ponto e vírgula.

- `string GetSegmentPoints(int id)`
  Retorna a coordenada dos pontos que discretizam o segmento cujo identificador foi enviado como parâmetro em “id” , sendo a latitude e longitude de cada ponto separados por ponto e vírgula, e cada coordenada separado por barra vertical.

- `string GetSegmentLength(int id)`
  Retorna o comprimento em metros do segmento cujo identificador foi enviado no parâmetro “id”.

- `string GetGPSByDateAndSegment(string date, int segmentid)`
  Retorna todas os registros de GPS que se referem a dados utilizados para monitorar um segmento. O registro contêm data e hora, ordem do ônibus, latitude e longitude separados por tabulação, sendo os registros separados por quebra de linha. O dia a ser considerado é enviado no parâmetro “date” em formato “aaaammdd” e o identificador do segmento em “segmentid”. O algoritmo para filtragem dos ônibus no segmento é o mesmo detalhado para o método que se segue.

- `string GetTrajetoriesByDateAndSegment(string date, int segmentid)`
  Retorna todas as passagens de ônibus pelo segmento cujo identificador é passado no parâmetro “segmentid”, para uma determinada data passada em “date”. O algoritmo de descoberta de trajetórias é apresentado no próximo capítulo. Essa passagem é organizada como se segue: ordem do ônibus que descreve a trajetória, latitude e longitude destes ônibus e

também a data/hora dos mesmos, total de metros, total de segundos e velocidade média da trajetória, índices dos ônibus em relação aos pontos que descretizam os segmentos, posição linear de cada ponto em relação ao segmento, intervalo e de metros entre cada ponto e velocidade média entre cada pontos. Os dados dos vetores são separados por tabulação, barra e ponto e vírgula nesta ordem de prioridade.

- `string TrafficSummaryDay(int segmentid, string date)`
  Retorna a consolidação das trajetórias do segmento cujo identificador é passado no parâmetro "segmentid", para uma determinada data passada em "date", visando o conhecimento do tráfego. Cada linha da *string* retornada representa um horário, com exceção da última que é o totalizador. As linhas possuem as quantidades totais de trajetos, registros de GPS dos ônibus, distância percorrida em quilômetros e tempo transcorrido em horas, assim como, velocidade média, nesta ordem, separados por tabulação.

- `string TrafficSummaryPeriod(int segmentid, string dates)`
  Retorna a estatística descritiva da consolidação das trajetórias do segmento cujo identificador é passado no parâmetro "segmentid", para datas determinadas, passadas em "dates", separadas por ponto e vírgula, visando o conhecimento do tráfego no período. Cada linha da *string* retornada representa um horário, com exceção da última que é o totalizador. As linhas possuem as quantidades totais médias de trajetos, registros de GPS dos ônibus, distância percorrida em quilômetros e tempo transcorrido em horas, assim como, velocidade média, nesta ordem, separados por tabulação.

- `string TrajectoriesNow(int segmentid)`
  Retorna as últimas trajetórias identificadas no segmento passado em "segmentid", da mesma forma que as trajetórias para uma data referencial são passadas, como explicado anteriormente.

- `string TrafficNow(int segmentid)`
  Retorna o tráfego atual no segmento passado em "segmentid", incluindo velocidade média, velocidade padrão, percentual relativo entre elas e atraso dos dados utilizados para o monitoramento em relação ao momento corrente.

- `string GetOverViewDay(string date)`
  Retorna a consolidação dos registros de GPS capturados na data enviada no parâmetro "date". Cada linha representa um horário, com exceção da última que é o totalizador. As linhas possuem o total de registros, registros sem informação de linha, linhas distintas, ônibus distintos, ônibus distintos operando e ônibus distintos operando sem linha, nesta ordem, separados por tabulação. A avaliação da situação do ônibus como operando ou não é feita considerando como referência o término de cada hora e desconsidera um mínimo de registros necessários, dado que ao longo de um hora espera-se obrigatoriamente um número de registros suficientes para o ônibus ser considerado operando durante a hora.

- `string GetOverViewPeriod(string dates)`
  Retorna a estatística descritiva da consolidação dos registros de GPS capturados nas datas determinadas, passadas em "dates", separadas por ponto e vírgula, visando conhecer informações gerais do serviço de transporte em questão. Cada linha representa um horário, com exceção da última que é o totalizador. As linhas possuem a média de registros, registros sem informação de linha, linhas distintas, ônibus distintos, ônibus distintos operando e ônibus distintos operando sem linha, nesta ordem, separados por tabulação. as quantidades totais médias de trajetos, registros de GPS dos ônibus, distância percorrida em quilômetros e tempo transcorrido em horas, assim como, velocidade média, nesta ordem, separados por tabulação.

Obviamente que novos métodos podem ser implementados neste serviço, para tornar outros conhecimentos disponíveis na Plataforma BusesInRio. Considerando que o acesso aos dados e as técnicas de manipulação destes estão implementadas em diversos serviços, criar um novo método tem seu desafio concentrado no algoritmo que se deseja desenvolver para extrair o conhecimento desejado, ficando transparente dificuldades de infraestrutura, capacidade computacional e outras que normalmente tornariam esse processo muito mais oneroso para o pesquisador.

Os conceitos que são utilizados em cada método, como "ônibus operando", "segmento/região", "passagem do ônibus pelo segmento", dentre outros, são explicados aos longos do próximo capítulo deste documento.

4.6 Soluções Clientes da Plataforma

Ao adotar o padrão WCF implementando SOA a Plataforma BusesInRio abre um vasto leque de ambientes e linguagens de programação para desenvolvimento dos clientes de seus serviços. No decorrer deste trabalho implementamos três clientes da Plataforma, a saber:

1) **Busesinrio.com** - Este Web Site foi desenvolvido em PHP, utilizando diretamente a API do Google Maps, e teve como objetivo primário permitir visualizações básicas dos dados armazenados, acessando diretamente os arquivos salvos, mas posteriormente foi adaptado para consumo dos serviços da Plataforma

BusesInRio. A seção 3.1 apresentou uma série de visualizações suportadas pelo Web Site.

2) **BusesInRio WinDemo** – Este *software* foi desenvolvido em C#.Net para execução em *desktop* com sistema operacional Windows, utilizando a biblioteca GMap.Net do MIT [78], tendo como objetivo demonstrar o consumo das visões históricas e em tempo real da Plataforma BusesInRio, assim como, os conceitos de região e segmento, permitindo a validação dos segmentos escolhidos. Detalhes deste *software* e dos conceitos são apresentados no próximo capítulo.

3) **BusesInRio App** – Esse aplicativo foi desenvolvimento em Swift para execução em dispositivos móveis (*smartphones*) com sistema operacional iOS, visando fornecer uma série de serviços ao usuário final, o cidadão/passageiro, a partir do consumo de conhecimentos da Plataforma BusesInRio. Esse aplicativo será detalhado em capítulo específico adiante.

4.7
Como utilizar a Plataforma em Pesquisas

Os serviços da Plataforma BusesInRio possuem acesso a um conjunto de bibliotecas pensadas para facilitar o desenvolvimento de novas pesquisas.

A biblioteca ***BlobMethods*** é uma das mais importantes, pois permite manipular os arquivos do *Blob Storage*. Seus métodos são:

- `public static CloudBlobContainer ConnectContainer(string storageConnectionString, string containerName)`
  Retorna a conexão um container do *Blob Storage*. Os dados da conta do *Blob Storage* devem ser enviados como parâmetro, assim como o nome do container desejado. A Plataforma utiliza como padrão os containers "csv", "seg", "count", "gps" e "gtfs", que podem ser entendidos como pastas. Outros containers podem ser criados para facilitar a organização de novas implementações.

- `public static void Download(CloudBlobContainer cloudBlobContainer, string blobname, string targetfile)`
  Obtêm um arquivo do *Blob Storage*. A conexão com o container e o nome do arquivo desejado dentre deste precisam ser enviados como parâmetro, assim como o local onde o arquivo obtido deve ser salvo.

- `public static void Upload(CloudBlobContainer cloudBlobContainer, string blobname, string sourcefile)`
  Envia um arquivo para o *Blob Storage*. A conexão com o container e o

nome a ser dado ao arquivo precisam ser enviados como parâmetro, assim como o local onde o arquivo que deve ser enviado se encontra neste momento.

- `public static void UploadTextToAppendBlob(CloudBlobContainer cloudBlobContainer, string blobname, string content)`
  Envia um texto para ser concatenado com os demais textos de um arquivo do *Blob Storage*. A conexão com o container e o nome do arquivo precisam ser enviados como parâmetro, assim como o texto a ser incluído.

- `public static void Delete(CloudBlobContainer cloudBlobContainer, string blobname)`
  Deleta um arquivo do *Blob Storage*. A conexão com o container e o nome do arquivo a ser excluído deste container devem ser enviados.

- `public static bool Exist(CloudBlobContainer cloudBlobContainer, string blobname)` Testa se um arquivo existe no *Blob Storage*. A conexão com o container e o nome do arquivo a ser verificada a existência no container devem ser enviados.

Cabe registro que `CloudBlobContainer` é um tipo para representar a conexão com o container do *Blob Storage* deo SDK do Azure do .Net, mais especificamente de *Microsoft.WindowsAzure.Storage.Blob*.

Outra biblioteca muito útil é a ***GisMethods***, que permite uma série de manipulações de dados georreferenciados, em especial:

- `public static double DistanceInMeters(GeographyPoint pointA, GeographyPoint pointB, bool absolute = true)`
  Testa se um arquivo existe no *Blob Storage*. A conexão com o container e o nome do arquivo a ser verificada a existência no container devem ser enviados.

- `public static GeographyPoint` TryFindPointInLine(`GeographyPoint` start, `GeographyPoint` end, `GeographyPoint` pt)
  Testa se um arquivo existe no *Blob Storage*. A conexão com o container e o nome do arquivo a ser verificada a existência no container devem ser enviados.

Cabe registro que `GeographyPoint` é um tipo para representar pontos georreferenciados de *System.Spatial* do .Net, podendo ser instanciado da seguinte forma:

- `GeographyPoint` point = `GeographyPoint`.Create(`double` lat, `double` lon)
  Instancia um ponto georreferenciados a partir de valores de latitude e longitude enviados como parâmetros.

Igualmente importante é a biblioteca ***Ionic.Zip*** incorporada na Plataforma BusesInRio, que oferece métodos para manipular arquivos compactados no formato ZIP. Sua utilização tende a ser combinada com *System.IO* e *System.Text*, do .Net, destinadas a manipulação de arquivos e textos.

A rotina apresentada a seguir ilustra um dos processos de maior interesse dos pesquisadores, o processamento de dados históricos de um determinado dia.

```
string blobname = "20160902";                              // nome do blob (arquivo)
string containername = "gps";                             // nome do container (pasta)
string storageConnectionString = "..."; // obtido na conta do Blob Storage do Azure
string file = @"D:\20160902.zip"; // caminho, nome e extensão do arquivo compactado
string txtName = "20160902.txt";         // nome e extensão do arquivo dentro deste
string separator = "\t";                           // separador dos campos do arquivo

CloudBlobContainer cloudBlobContainer =  BlobMethods.ConnectContainer(
   storageConnectionString, containername);                  // conecta ao container
BlobMethods.Download(cloudBlobContainer, blobname, file);       // download do blob
if(!File.Exist(file)) return;               // testa se arquivo foi criado (existe)
using (ZipFile zip = ZipFile.Read(file))                  // abre arquivo compactado
{
   ZipEntry zipEntry = zip[txtName];        // abre arquivo de dentro do compactado
   using (StreamReader read = new StreamReader(zipEntry.OpenReader()))
   {
      while (read.Peek() >= 0) // continua se ainda existem linhas para serem lidas
         if (text.Substring(0, 1) == "#") continue;    // testa se é linha de dados
         string text = read.ReadLine();                                // lê uma linha
         string[] fields = text.Split(separator);  // separa os campos de uma linha
         try {
            DateTime datetime = DateTime.ParseExact(
              fields[0], "dd-MM-yyyy HH:mm:ss",
              CultureInfo.InvariantCulture);            // obtêm data e hora do gps
              string ordem = fields[1];    // obtêm ordem (identificador) do ônibus
              string linha = fields[2];                    // obtêm linha do ônibus
              double lat = 0;
              double lon = 0;
              Double.TryParse(fields[3], out lat);
              Double.TryParse(fields[4], out lon);
              GeographyPoint point =
                 GeographyPoint.Create(lat, lon);        // obtêm posição do ônibus
              . . .                           // implementa o que desejar com os dados
        } catch() { }
      }
   }
}
File.Delete(file);                                    // deleta o arquivo compactado
```

Figura 16 – Rotina para processar arquivos de texto compactados

A fim de não estender demasiadamente esta seção, referenciamos o repositório da Plataforma BusesInRio, onde o código fonte de cada um dos serviços pode ser obtido e estudado em maior profundidade, junto com sua documentação adicional. Acesse: https://busesinrio.codeplex.com/.

Na seção 5.12 será apresentado um exemplo de implantação de extração de conhecimento utilizando a Plataforma BusesInRio, porém, recomendamos a leitura em ordem, para que os conceitos envolvidos estejam compreendidos.

5
Extraindo Conhecimento dos Dados sobre o Tráfego

5.1
Seleção de Segmentos

Em uma visão mais abrangente, os ônibus podem ser considerados sensores que viabilizam a compreensão dos padrões e identificação de anomalias no tráfego de veículos nas áreas urbanas. Naturalmente, a cobertura destes sensores se concentra nas principais ruas da cidade, uma vez que os trajetos das linhas de ônibus utilizam prioritariamente vias de transito rápido e arteriais de maior circulação, evitando em seu planejamento vias menores, como locais e coletoras. Além disso, podemos imaginar que algumas ruas principais da cidade não são cobertas por nenhuma linha de ônibus.

A escolha dos segmentos da cidade que serão monitorados, entendendo segmento como uma rua, avenida ou parte destas, é fundamental para se extrair conhecimento sobre o tráfego da cidade a partir destes dados de GPS dos ônibus.

Nessa seção, apresentaremos a abordagem adotada para validação de um segmento, para que o mesmo passe a ser monitorado. Essa abordagem poderia ser repetida para todas as ruas/avenidas da cidade, para uma malha de segmentos mais representativa possível da área urbana (estratégia dita como "gulosa"). A abordagem consiste nos passos abaixo detalhados:

PARTE I – Preparação dos dados sobre a cidade

1) Download de mais de 300 MB de informações do OpenStreetMap [80] sobre o mapa da Cidade do Rio de Janeiro.
2) Extração apenas das informações necessárias e organização em dois arquivos, utilizando a ferramenta de QGIS [82], conforme se segue.

Campo	Descrição
Id	Identificador único do polígono que representa parte do mapa da cidade
Nome	Nome da rua ou avenida que o polígono representa

Tabela 15 – Campos do arquivo poligonos-nomes.txt

Campo	Descrição
Id	Identificador do polígono que representa parte do mapa da cidade
Latitude	Latitude de uma coordenada do polígono (GPS, WGS84)
Longitude	Longitude de uma coordenada do polígono (GPS, WGS84)

Tabela 16 – Campos do arquivo poligonos-coordenadas.txt

PARTE II – Delimitando a região onde o segmento está contido

3) A partir do nome da rua ou avenida contido no arquivo "poligonos-nomes.txt" (veja Tabela 15) é possível selecionar os identificadores dos polígonos candidatos a descrever o segmento com este nome e, consequentemente, suas coordenadas no arquivo "poligonos-coordenadas.txt" (veja Tabela 16). Em ambos os casos, uma busca sequencial, com comparação textual linha a linha, é suficientemente eficiente. A Figura 17 ilustra a projeção desses dados no *GoogleMaps* para o nome "Rua Jardim Botânico". Desenvolvemos uma ferramenta para *desktop* (em C# .Net) específica para essa visualização, que acrescenta um marcador verde na primeira coordenada do polígono e vermelho na última, sem a intenção de indicar qualquer sentido neste caso. Além disso, ela liga as coordenadas do polígono com linhas para facilitar a visualização. Complementarmente, ao clicar em um marcador é apresentado o identificador do polígono.

Figura 17 – Discretização OpenStreetMap da "Rua Jardim Botânico"

4) Como os dados do OpenStreetMap são cadastrados de forma colaborativa pelos próprios usuários, é possível que polígonos incorretos seja desenhados no mapa. Sendo assim, é essencial confrontar visualmente esses polígonos do *OpenStreetMap* com um

mapa de outra origem, como é feito na Figura 17 com o GoogleMaps. Ao clicar nos marcadores deste polígonos é possível obter seu identificador, conforme ilustrado na Figura 18. Esses identificadores podem ser inseridos em uma lista de polígonos a serem desconsiderados no passo anterior.

Figura 18 – Visualização de polígono inconsistente do OpenStreetMap

5) Considerando apenas os polígonos validados é possível utilizar a latitude e longitude de menor e maior valor dentre todas as coordenadas dos polígonos para escolher os pontos que definem a menor região retangular que contenha todos estes polígonos (*bounding box*), como ilustrado na Figura 19.
6) Deste ponto em diante, passamos a considerar o nome utilizado como o prefixo do segmento a ser monitorado e a região caracterizadas pelas suas latitudes e longitudes mínimas e máximas como seus limites. Com base no exemplo utilizado anteriormente, teríamos então o segmento "Rua Jardim Botânico" com sua região caracterizada pelos valores: 22.974896;-22.9603236;-43.2263832;-43.2045122.

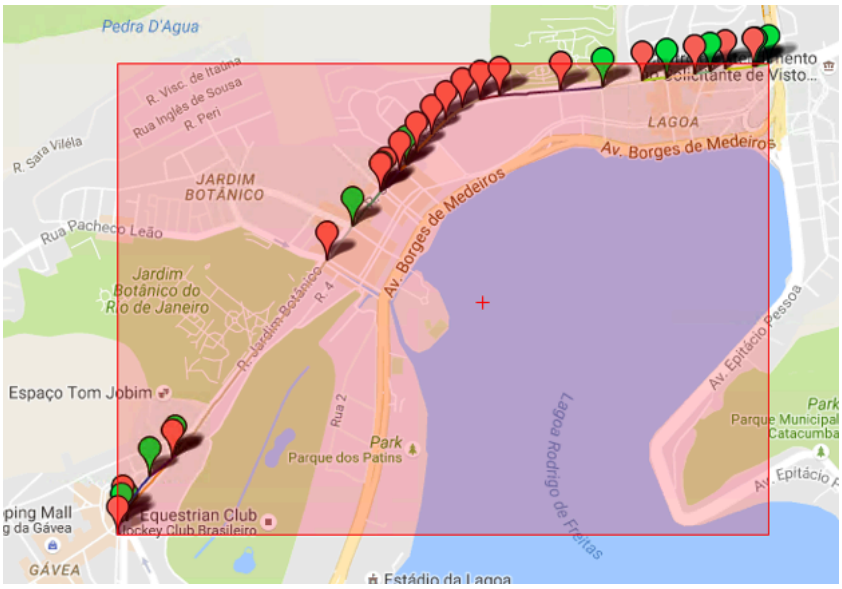


Figura 19 – Região que delimita a "Rua Jardim Botânico"

PARTE III – Busca trajetórias de linhas de ônibus da região

7) Utilizando uma coordenada central dos polígonos validados anteriormente para a região (a coordenada central é o ponto – latitude/longitude – dentre os que descrevem o polígono, que está mais próximo da metade do seu comprimento, o que será abordado em momento oportuno deste trabalho), é possível selecionar dentre todas as trajetórias de linhas de ônibus conhecidas quais tem uma coordenada próxima desta, entendendo como próxima uma distância linear de 100 metros ou menos entre as coordenadas (coordenada central dos polígonos da região x uma coordenada que discretiza uma trajetória de uma linha de ônibus). O valor de 100 metros refere-se ao erro típico de dispositivos de GPS a ser tratado na seção 5.8. As trajetórias correntes da linhas de ônibus são disponibilizadas por métodos apresentados na seção 3.3 deste trabalho, não cabendo aprofundamento nessa seção. A Figura 20 ilustra as trajetórias com proximidade a coordenada do segmento "Rua Jardim Botânico", utilizando a mesma ferramenta de visualização das figuras anteriores.
8) Se nenhuma trajetória for encontrada, o segmento em questão não é uma boa escolha para ser monitorado, dado que a probabilidade de não existirem sensores no mesmo é alta, ou seja, de não existirem ônibus passando pelo segmento. Caso contrário, podemos seguir a análise

considerando apenas as coordenadas das trajetórias que estão contidas dentro da região do segmento. A Figura 21 ilustra para o exemplo "Rua Jardim Botânico" a região com as trajetórias contidas. Cabe observar que como existem coordenadas de início e fim muito próximas, existe superposição de marcadores, o que dificulta a visualização nessa imagem estática (na figura anterior é fácil ver o marcador de início de oito trajetórias, mas nessa figura só se destacam dois, apesar de na realidade existirem todos os oito, pois eles passam na região).

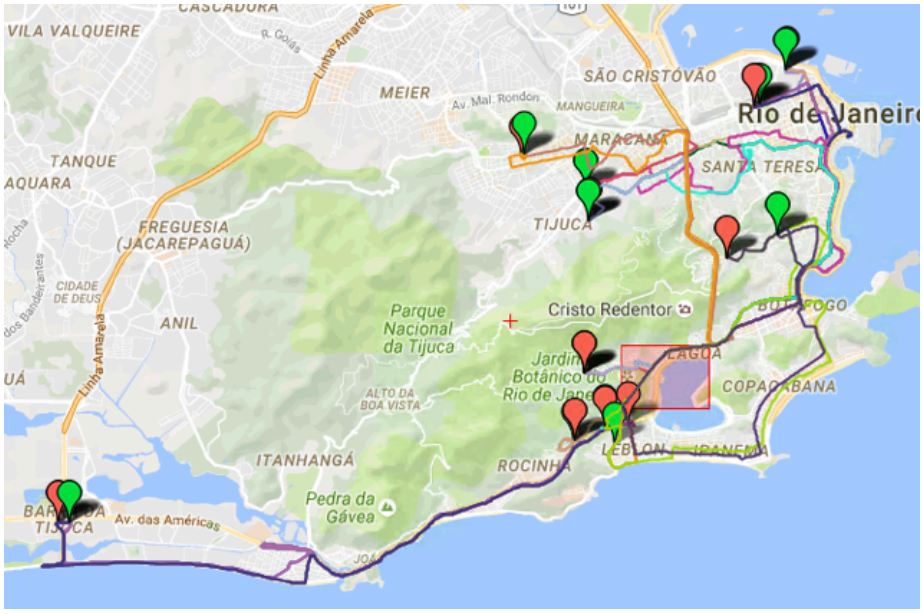


Figura 20 – Trajetórias que passam pelo centro do segmento

Figura 21 – Trajetórias dentro da região do “Rua Jardim Botânico”

9) O passo seguinte é trivial, a escolha da trajetória que representa o segmento. Primeiramente temos que manter em mente os polígonos que deram origem ao segmento (veja Figura 19), como ilustrado anteriormente no item 5. Esse cuidado evita a escolha de uma trajetória que não apresenta o mesmo início e fim do segmento original, como é o caso da que se inicia no topo esquerdo superior da Figura 21. Clicando nos marcadores de início é possível visualizar o identificador da trajetória e selecioná-la.

10) Em especial, quando existirem trajetórias iniciando em sentidos opostos do segmento, podemos deduzir que o monitoramento deste segmento pode ocorrer nos seus dois sentidos. Sendo assim, cabe dividir o segmento em dois, mantendo a região e seu prefixo, acrescido de um sufixo em seu nome para representar o sentido. Mais uma vez, utilizando o segmento “Rua Jardim Botânico”, podemos escolher as trajetórias ilustradas na Figura 22 e na Figura 23 para representar seus dois sentidos.

Figura 22 – Segmento “Rua Jardim Botânico (Humaitá => Gávea)”

Figura 23 – Segmento “Rua Jardim Botânico (Gávea =>Humaitá)”

PARTE IV – Arquivo de segmentos da cidade

11) Repetindo a PARTE II e III para as ruas/avenidas desejadas formamos um arquivo com os segmentos a serem monitorados na cidade, conforme se segue. A Tabela 17 lista os campos contidos num arquivo do tipo “segmentos.txt”. Destacamos que as Coordenadas citadas abaixo referem-se a geometria do segmento a ser monitorado.

Campo	Descrição
Id	Identificador único sequencial do segmento
Nome	Nome do segmento incluindo, se necessário, seu sentido
Região	“Lat. Mín. ; Lat. Máx. ; Long. Mín. ; Long. Máx.” da região
Coordenadas	“Lat.;Lon.|Lat.;Lon.|Lat.;Lon.|Lat.;Lon.|....” do segmento

Tabela 17 – Campos do arquivo segmentos.txt

A figura da próxima página apresenta os dados para o segmento utilizado como exemplo, que estarão contidos neste arquivo acima mencionado, uma vez que estes dois segmentos foram escolhidos.

Conforme mencionado anteriormente, essa abordagem poderia ser repetida para todas as ruas/avenidas da cidade, ou seja, todos os nomes distintos do arquivo “poligonos-nomes.txt”, para uma malha de segmentos mais representativa possível da cidade. No entanto, no âmbito deste trabalho tal completude não se justifica, portanto, optamos pela execução desta abordagem para as ruas/avenidas que entendemos representar de forma geral o estado do tráfego da Zona Sul da Cidade do Rio de Janeiro, área que despertava interesse do autor.

O subconjunto manualmente escolhido foi:

1. Rua Jardim Botânico
2. Rua São Clemente
3. Rua Voluntários da Pátria
4. Rua Barata Ribeiro
5. Avenida das Américas
6. Nossa Senhora de Copacabana

```
Rua Jardim Botânico (Gávea => Humaitá)
-22.974896;-22.9603236;-43.2263832;-43.2045122
-22.9748561955569;-43.2259347147153|-22.974533505317;-43.226115393011|-22.9744387176726;-43.2261323092586|-22.9742678707656;-43.2261041397942|-22.974052582060 9;-
43.2259433787839|-22.9738707913039;-43.2258378631269|-22.9736475382033;-43.225658524821|-22.9731454071348;-43.2251713521761|-22.97265684144;-43.224685836082 4|-
22.9724426283246;-43.2245076021441|-22.97229198109;-43.2243608426787|-22.9697801638234;-43.2222261857898|-22.9663799413811;-43.2193870647612|-22.965336706224;-
43.2185638829168|-22.964761571789;-43.2180106748527|-22.9641912523244;-43.2174842306127|-22.9641136678115;-43.2174108891012|-22.9635751402805;-43.216901095808 9|-
22.96340615497;-43.2167302156269|-22.963000074194;-43.2163546092777|-22.9625071126131;-43.2158820173412|-22.9621600561416;-43.2155355040602|-22.9620319971789;-
43.215390297724|-22.9619358236793;-43.2152702724051|-22.9618210342133;-43.2151005118791|-22.9616692412424;-43.2148526146564|-22.9616042949023;-43.2146986579167|-
22.9615528500839;-43.2145408673671|-22.9615065044216;-43.2144348157141|-22.9614132815976;-43.2141757893812|-22.9613566908586;-43.2139619926518|-22.9612814904484;-
43.2135971835631|-22.9611741833418;-43.2126742920848|-22.9610447124435;-43.2115607659165|-22.9608805528952;-43.2101489067466|-22.960810186341;-43.2095437160696|-
22.9607384618105;-43.2087901210672|-22.9606523778049;-43.2079647988364|-22.9605511330283;-43.2069970581676|-22.9604928920333;-43.2065089529258|-22.9604364092129;-
43.2060355929682|-22.9603397302943;-43.2050732797081
Rua Jardim Botânico (Humaitá => Gávea)
-22.974896;-22.9603236;-43.2263832;-43.2045122
-22.9603397302943;-43.2050732797081|-22.9604364092129;-43.2060355929682|-22.9604928920333;-43.2065089529258|-22.9605511330283;-43.2069970581676|-22.9606523778049;-
43.2079647988364|-22.9607384618105;-43.2087901210672|-22.960810186341;-43.2095437160696|-22.9608805528952;-43.2101489067466|-22.9610447124435;-43.2115607659165|-
22.9611741833418;-43.2126742920848|-22.9612814904484;-43.2135971835631|-22.9613566908586;-43.2139619926518|-22.9614458586394;-43.214261907561|-22.9615065044216;-
43.2144348157141|-22.9615528500839;-43.2145408673671|-22.9616042949023;-43.2146986579167|-22.9616692412424;-43.2148526146564|-22.9618210342133;-43.2151005118791|-
22.9619358236793;-43.2152702724051|-22.9620319971789;-43.215390297724|-22.9621600561416;-43.2155355040602|-22.9625071126131;-43.2158820173412|-22.963000074194;-
43.2163546092777|-22.96340615497;-43.2167302156269|-22.9635751402805;-43.2169010958089|-22.9641136678115;-43.2174108891012|-22.964236911804;-43.2175293433045|-
22.964761571789;-43.2180106748527|-22.965336706224;-43.2185638829168|-22.9664893511026;-43.2194801059221|-22.9697801638234;-43.2222261857898|-22.97229198109;-
43.2243608426787|-22.9724426283246;-43.2245076021441|-22.9726903061127;-43.2248518032681|-22.9730699711022;-43.2252893720069|-22.9733990789043;-43.225647749144 3|-
22.9737684376632;-43.2259714709066|-22.9741816239827;-43.2261769785846|-22.9742861818618;-43.2262402270048
```

Figura 24 – Exemplo de dados de “segmentos.txt”

5.2
Checagem do Segmento Selecionado

Uma vez selecionados os segmentos para monitoramento do tráfego da cidade do Rio de Janeiro, precisamos validar se uma quantidade relevante de ônibus passam nesse segmento ao longo do dia, ou seja, se temos sensores suficientes para monitorar o segmento. Para tanto, desenvolvemos um *software* em C#.Net para execução em *desktop* com sistema operacional Windows, utilizando a biblioteca [78], que apelidamos de "BusesInRio WinDemo".

Esse *software* tem como objetivo demonstrar o consumo das visões históricas e em tempo real da Plataforma BusesInRio, assim como, os conceitos de região e segmento, permitindo a validação dos segmentos escolhidos.

A Figura 25 facilita a explicação do funcionamento deste *software*. Sua interface se encontra em Inglês, dado que o mesmo foi utilizado para seção de demonstração de artigo apresentado em [75].

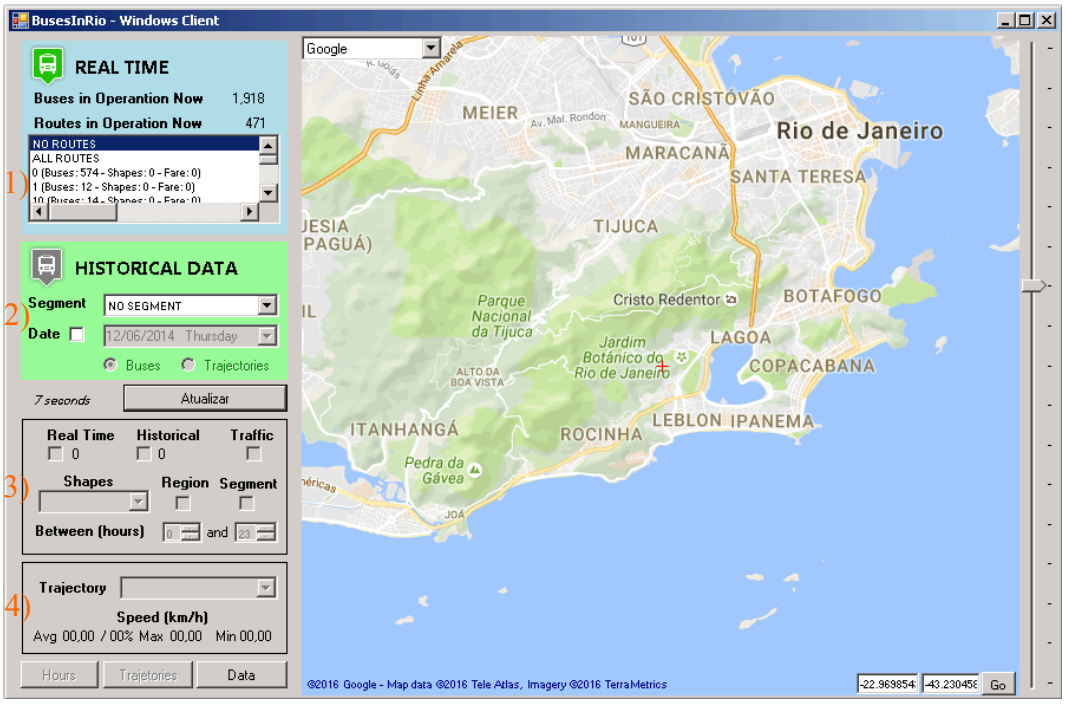


Figura 25 – Tela inicial do *software* BusesInRio WinDemo

Na esquerda da interface, de cima para baixo, temos:

a) Na primeira área (número 1), o número de ônibus e linhas operando na cidade no instante em que a última consulta foi realizada. Esse número é obtido com uma simples consulta aos métodos

`HowManyBusesOperating` e `HowManyRoutesOperating` da Plataforma BusesInRio. São essas informações da situação corrente (“tempo quase real”).

b) Na mesma área (número 1), todas as linhas de ônibus da cidade listadas, independente se possuem ou não ônibus operando, com diversas informações, como seu identificador, nome, número de ônibus operando, número de trajetórias conhecidas e tarifa. Esses dados são obtidos com uma simples consulta ao método `GetRoutesStatus` da Platforma, passando o valor 2 como parâmetro.

c) Na outra área (número 2), temos os segmentos monitorados da cidade listados pelo seu nome. Esses dados são obtidos com uma simples consulta ao método `GetSegments` da Plataforma BusesInRio. Abaixo deste, ainda na mesma área, existe um campo para se filtrar dados históricos por uma data específica.

d) A área seguinte (número 3), destina-se a manipular no mapa a exibição dos dados obtidos, ou seja, ocultar e exibir informações, de acordo com filtros.

e) Por último (área número 4), são exibidas informações consolidadas sobre os dados obtidos com os filtros, que serão explicados melhor mais adiante.

A Figura 26 apresenta o resultado da seleção de uma linha de ônibus e sua exibição no mapa no lado direito. Repare que linhas coloridas são formadas para representar os trajetos desta linha, obtidos com a conexão dos pontos que discretizam cada um dos seus trajetos. Também é apresentado no mapa a última coordenada de ônibus que declara pertencer a linha em questão. Esses dados podem ser obtidos utilizando os métodos `GetBusesFromRoute`, `GetShapesId` e `GetShape` da Plataforma, a partir do identificador da linha de ônibus selecionada.

Na parte do mapa é possível ainda alterar o mapa a ser utilizado entre o Google, Bing, OpenStreetMap e outros provedores de mapa, o que permite dirimir qualquer dúvida entre as projeções, assim como, é possível mudar o zoom aplicado ou o ponto central do mapa.

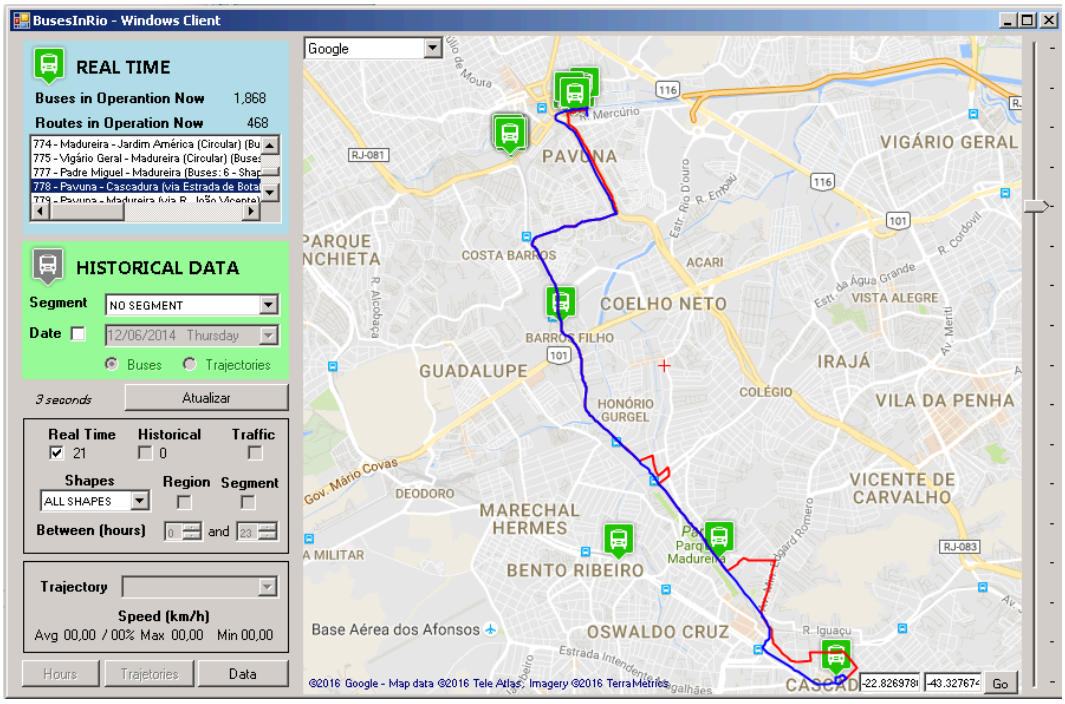


Figura 26 – Visualização de uma linha de ônibus no WinDemo

Na área número 4 (explicada na letra "d" anteriormente), podemos, por exemplo, para uma determinada linha de ônibus, escolher apenas uma de suas trajetórias para visualização. Para ilustrar o uso de diferentes mapas, utilizamos o OpenStreetMap na Figura 27 (alterando *Shapes* de *ALL SHAPES* para o número de um dentre os listados, seria visualizado apenas a rota escolhida).

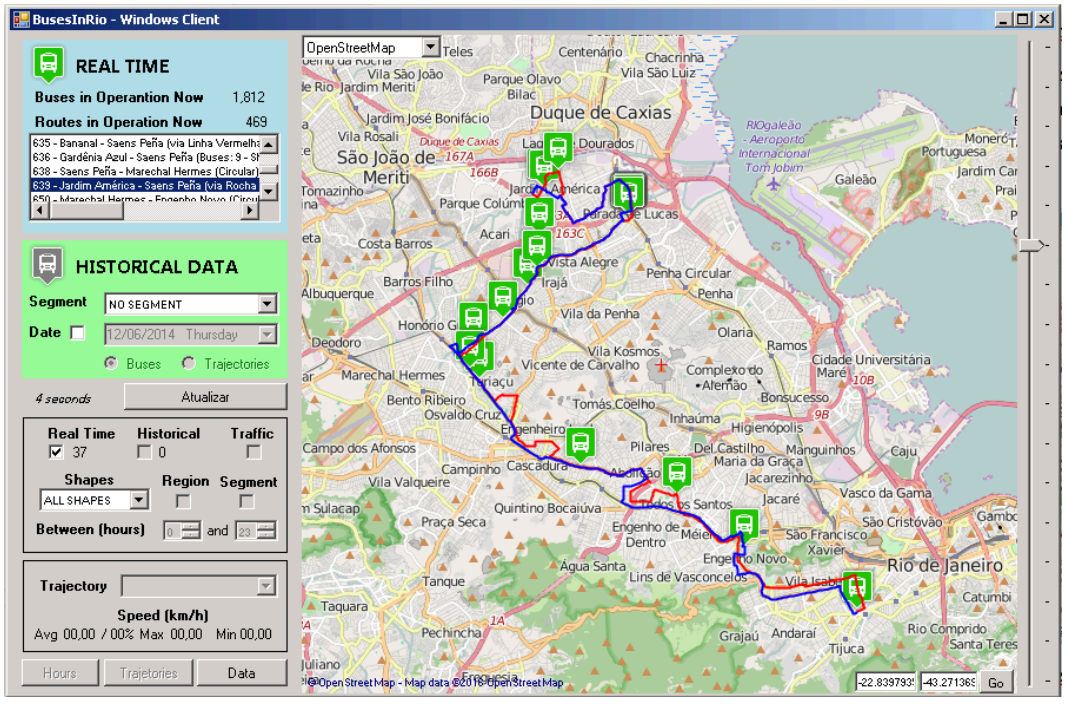


Figura 27 – Visualização de uma trajetória no WinDemo

Retomando a questão central desta seção, podemos escolher um segmento no lado esquerdo e visualizá-lo no mapa. A Figura 28 apresenta o segmento "Rua Jardim Botânico (Gávea=>Humaitá)" no mapa, exibindo tanto sua região em vermelho, quanto sua discritização conectada em verde. Esses dados foram obtidos dos métodos `GetSegmentRegion` e `GetSegmentPoints` da Plataforma.

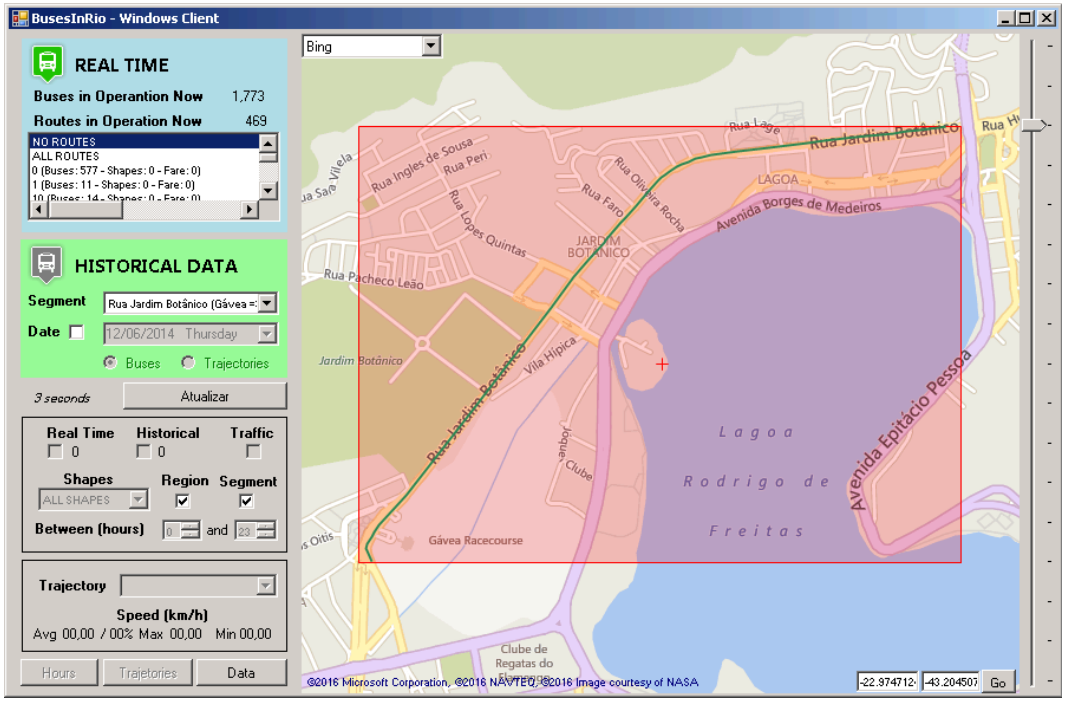


Figura 28 – Visualização de um segmento no WinDemo

A Figura 29 mantêm o campo *Historical* selecionado (i.e. informa que deseja exibir todos os registros no mapa), ou seja, todos os ônibus que passaram pelo segmento na data escolhida, no caso, 09/06/2016, serão apresentados no mapa. Para tornar mais agradável a visualização, a região não é apresentada no mapa (campo "Region" está desmarcado"). O segmento é apresentado (campo "Segment" está marcado), mas como o volume de ônibus é enorme, o segmento é completamente coberto. Destacamos que os ônibus históricos são apresentados em cinza, diferentes dos recentes, que foram apresentados em verde. Esses dados foram obtidos do método `GetGPSByDateAndSegment` da Plataforma BusesInRio. Fica evidente nesta imagem, pela explosão de dados de GPS dos ônibus, que o segmento aparentemente possui sensores suficientes ao longo do dia para que o mesmo possa ser monitorado.

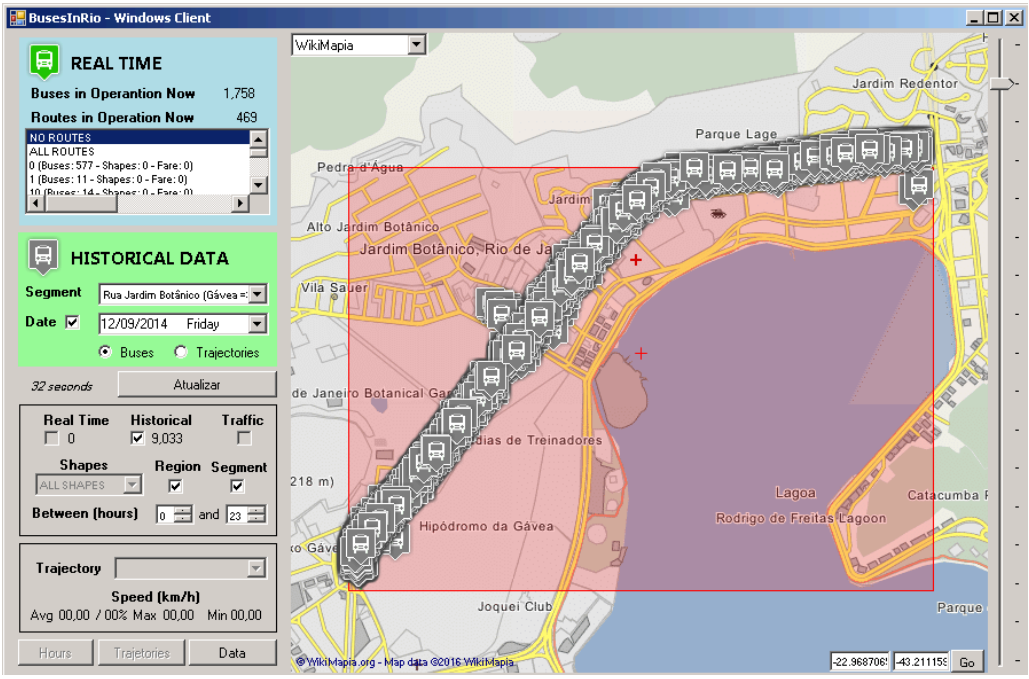


Figura 29 – Dados históricos de GPS em um segmento no WinDemo

Aplicando diferentes filtros por horário, podemos validar se existem dados suficientes para o monitoramento do segmento ao longo de todo o dia. Ao longo dessas manipulações, o número de ônibus (densidade) é apresentada na área de número 3. A Figura 30 apresenta os dados anteriores filtrados pelo horário de 23 às 23 horas.

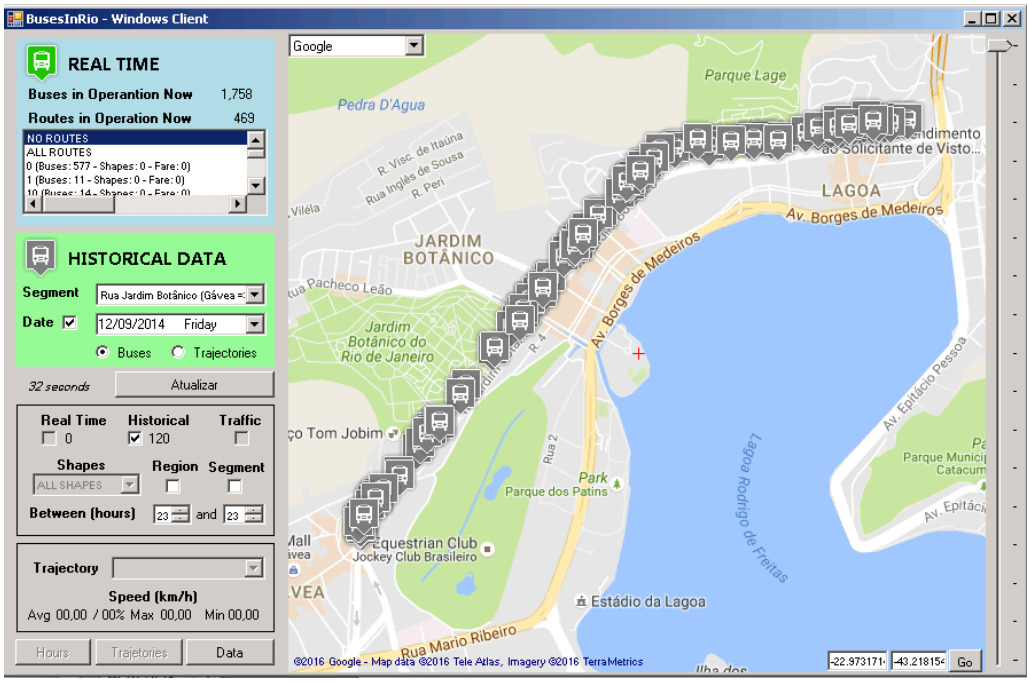


Figura 30 – Dados históricos de GPS em uma hora no WinDemo

A simples visualização de cada segmento selecionado na seção 5.1 permite validar estas escolhas. Naturalmente, como escolhemos ruas sabidamente de grande cobertura pelas linhas de ônibus, todas tendem a ser validadas, ou seja, ônibus suficientes passam nessas ruas ao longo de todo dia. No caso específico, todas as listadas no final da seção anterior foram validadas.

Cabe complementar a apresentação do *software* explicando seus botões:

a) O botão "Update" ("Atualizar") permite realizar as consultas a Plataforma BusesInRio com os parâmetros selecionados, atualizando a visualização no mapa.
b) O botão "Data" salva todas as informações recebidas da Plataforma BusesInRio em um arquivo texto para validação das visualizações no mapa e para análises complementares que forem desejadas.
c) O botão "Trajctories" apresenta uma análise da passagem dos ônibus pelo segmento, conforme apresentaremos ainda neste capítulo.
d) A última caixa, também consolida informações sobre essa passagem. Os conceitos, algoritmos e visões relativas a essa análise são tratadas na próxima seção.

5.3 Identificação das Trajetórias dos Ônibus pelo Segmento

Uma vez selecionados e validados os segmentos para monitoramento do tráfego da cidade do Rio de Janeiro, podemos fazer uso dos dados históricos salvos na Plataforma BusesInRio para identificar os padrões de tráfego deste segmento, considerando diferentes dias, horários ou situações. Essa abordagem foi inicialmente debatida em [76]. Sendo implementado um algoritmo de extração de trajetórias como se segue, aprimorado após exaustivas testagens.

1) Para cada data, em ordem cronológica, da mais antiga para a mais recente (ano aaaa, mês mm e dia dd) é copiado o arquivo "storage\gps\aaaammdd.zip" para a instância que processará a requisição.
2) O arquivo é descompactado e lido linha a linha, para cada segmento a ser monitorado (id do segmento), sendo aplicado o algoritmo descrito pelo diagrama da Figura 31 para cada registro (linha).

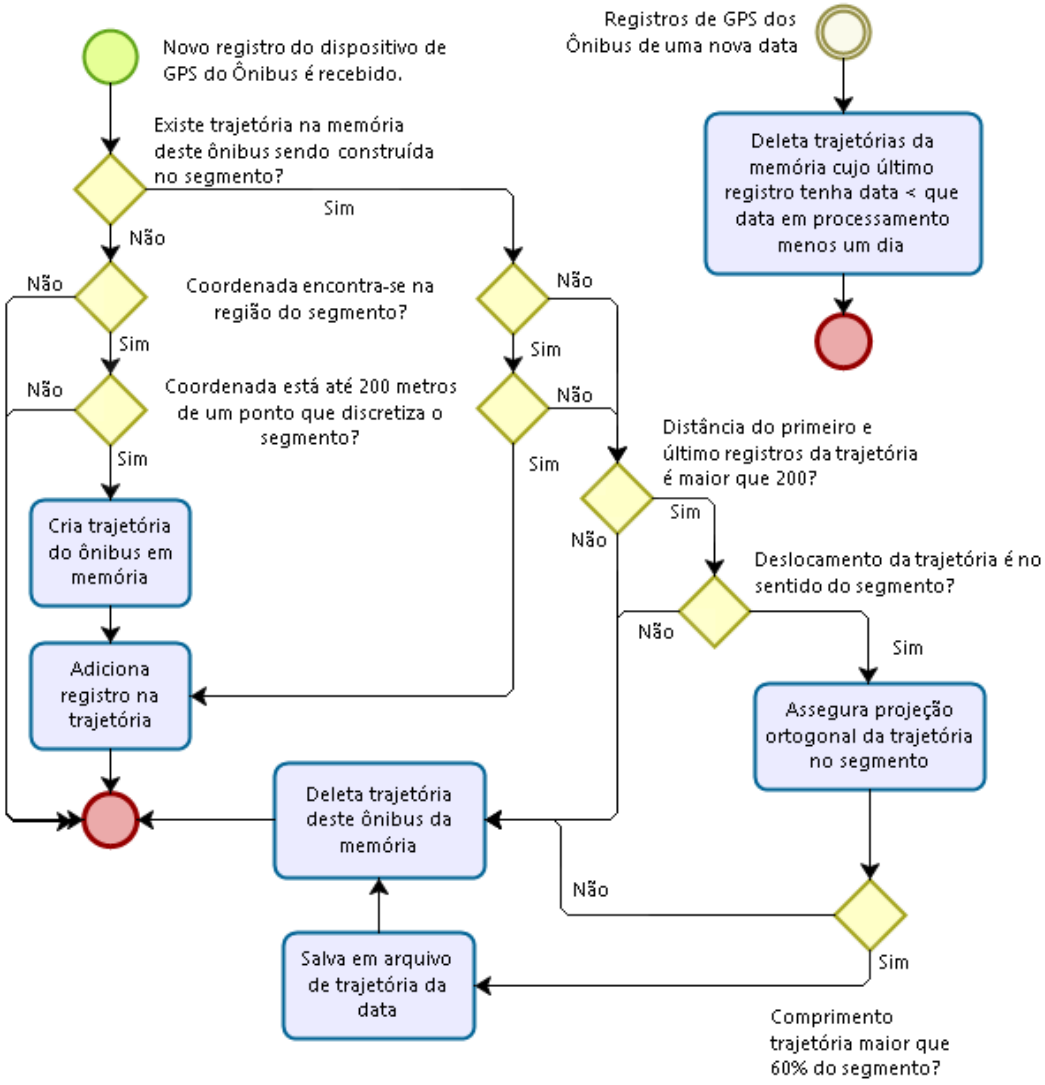


Figura 31 – Diagrama do algoritmo de descoberta de trajetórias

3) Ao final da extração das trajetórias do segmento para data, o arquivo gerado na instância é salvo no armazenamento da Plataforma BusesInRio na pasta "seg" com o formato ("[yyyymmdd]_seg[id].txt"). As trajetórias do dia anterior, são incluídas no arquivo respectivo. As trajetórias não concluídas permanecem na memória.
4) Ao final do dia (troca de data), todas as trajetórias que não tenham o último ponto no dia anterior ao que será processado são excluídas, pois não existirá dado suficiente para concluí-las. Repare que algumas trajetórias podem ser iniciados em uma data e concluídos no dia seguinte, além, obviamente das concluídas no mesmo dia. Esse processo evita que falhas na informação dos dados pela Prefeitura resulte em

trajetórias inválidas. Esse processamento de limpeza é apresentado no lado superior do diagrama.

Cabe registro que esta abordagem onde seguimos a trajetória do ônibus considerando cada ponto sobre o segmento, permite detectar o que acontece entre o ponto de entrada e saída do segmento, evitando que situações que ocorrem entre esses pontos sejam ignoradas, como eventuais desvios ou interrupções no *stream* de dados.

O método `GetTrajetoriesByDateAndSegment` da Plataforma BusesInRio permite o acesso ao resultado do algoritmo apresentado. Além disso, os resultados deste método podem ser explorados e até consolidados no software supracitado, conforme imagem da Figura 32.

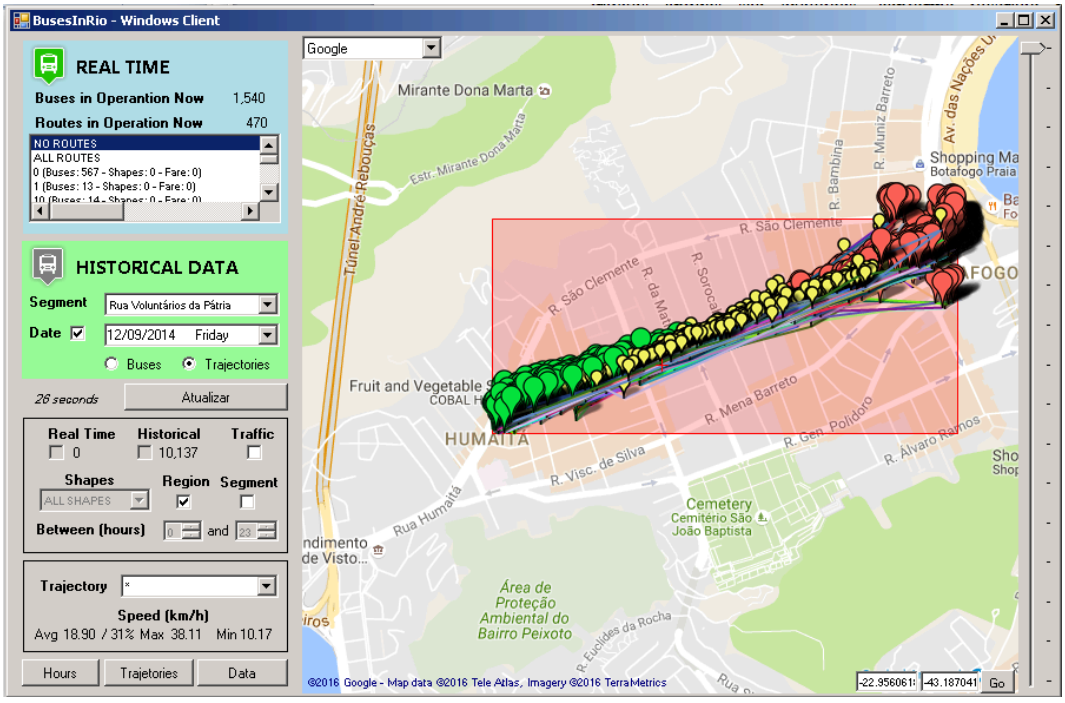


Figura 32 – Trajetórias da data sobre um segmento no WinDemo

Nessa representação visual podemos entender cada ponto em verde como o primeiro registro de um ônibus no trajeto no segmento e em vermelho como o último registro de um ônibus, portanto, sinalizam o início e fim de um trajeto. Os pontos em amarelo são registros intermediários deste trajeto. A conexão entre os pontos é apenas para facilitar a visualização, e esta adota diferentes cores.

Cabe observar que a quantidade de segmentos para uma data é considerável, tornando a visualização extremamente densa. No entanto, o software permite

filtrar as trajetórias individualmente ou pelo horário de início, confirme ilustrado na Figura 33, para o horário de 10 horas da manhã e posteriormente na Figura 34 para uma trajetória em especial.

Esses filtros foram de suma importância para depuração visual e aprimoramento do algoritmo supracitado. Além disso, as funcionalidades implementadas nesse software podem ser reutilizadas para interações diversas com a Plataforma, criando assim um ambiente propício aos novos pesquisadores.

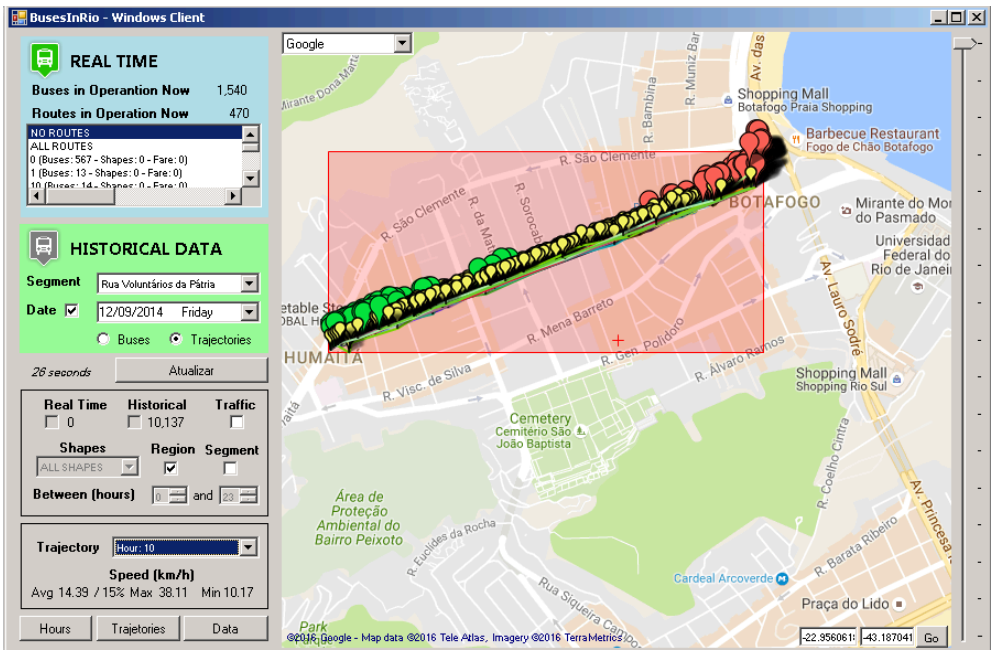


Figura 33 – Trajetórias na data/segmento iniciadas as 10h

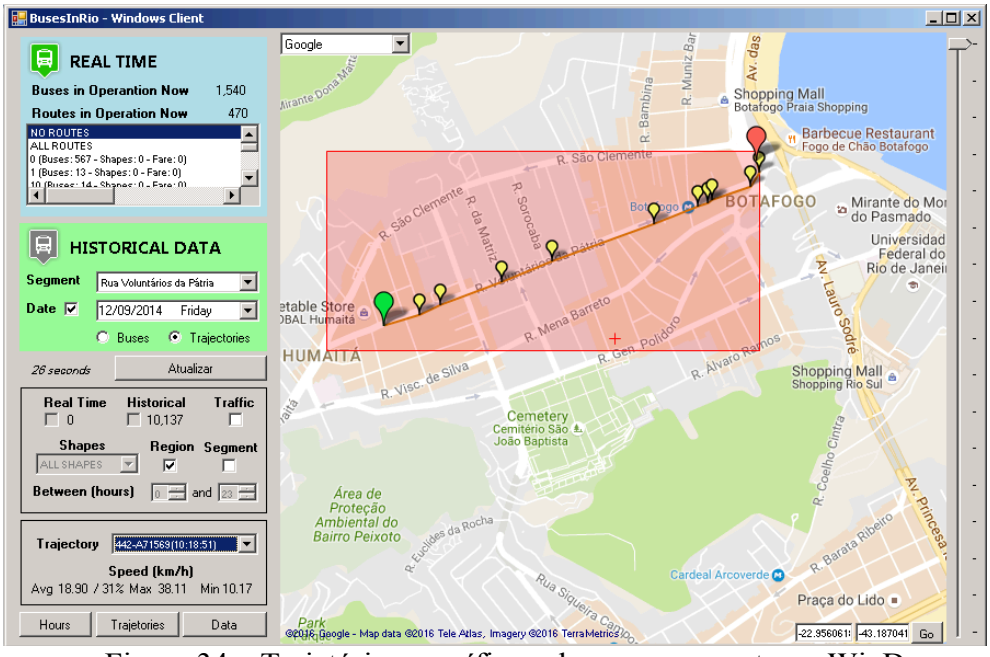


Figura 34 – Trajetória específica sobre um segmento no WinDemo

A seguir apresentamos uma trajetória individualmente, ocorrida em um segmento e data específicos, conforme consta na Figura 35, escolhida especialmente para facilitar a compreensão de questões relevantes, como o cálculo de distância.

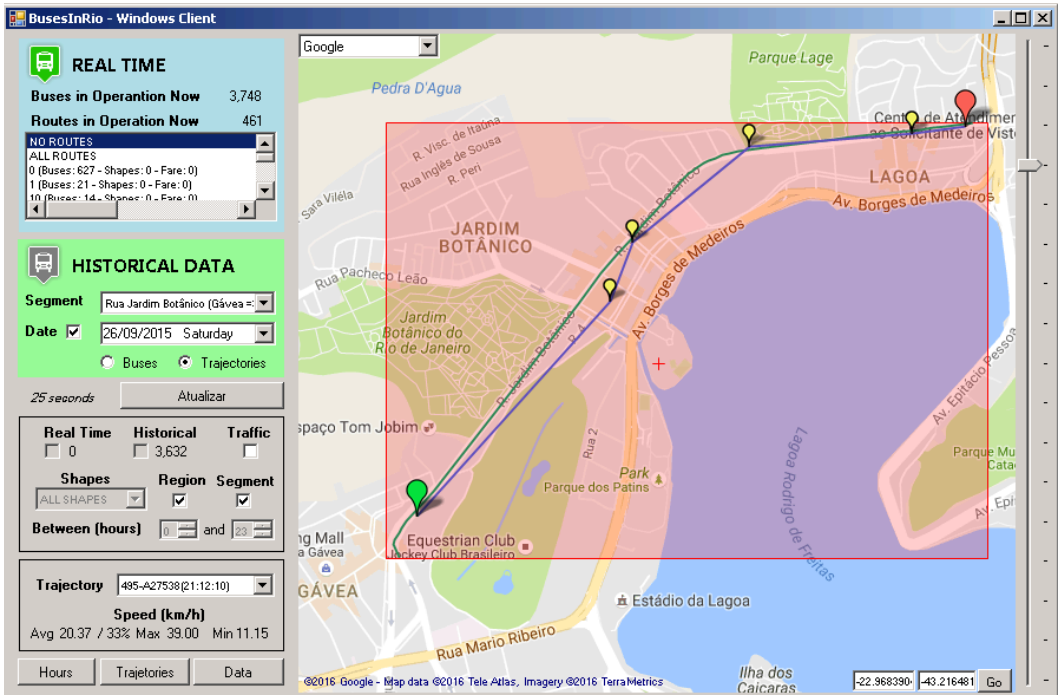


Figura 35 – Trajetória no WinDemo ilustrando uma situação

Nessa figura mantivemos a apresentação do segmento (em verde) e da região (zona vermelha). Repare que o segmento em questão apresenta uma curva acentuada e os registros capturados ocorreram antes e depois desta curva. Com isso, a distância entre os pontos, uma linha reta, é menor que a curva. Isso ocorre em função do segmento apresentar uma discretização mais detalhada.

Outra questão interessante ocorre no segundo registro da trajetória, logo após o inicio (verde). Este registro informa que o ônibus naquele momento estava dentro de um prédio, como é possível visualizar. Obviamente que trata-se de alguma interferência na precisão do dispositivo de GPS do ônibus, que na verdade estava percorrendo a rua do segmento, como fica evidente ao analisar a trajetória como um todo.

Dessa forma, fica claro que calcular a distância entre dois registros não representa a distância percorrida pelo ônibus naquele tempo. Nesse sentido, se faz necessário lançar mão de uma definição importante da geometria.

A projeção ortogonal de um ponto P sobre uma reta r é o ponto P' de r que está mais próximo de P. Dessa forma, P' é o ponto de interseção entre a reta r e a reta s que passa por P e é perpendicular a r, conforme ilustrado na Figura 36.

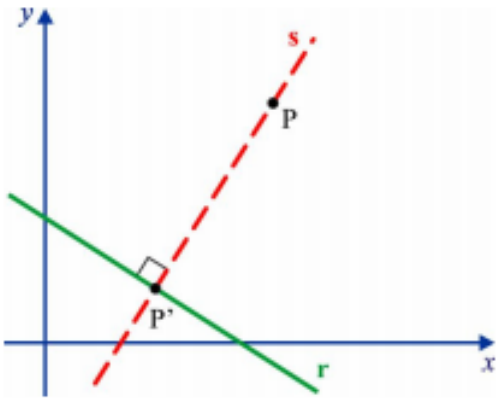


Figura 36 – Projeção ortogonal

Considerando que o segmento é discretizado por uma sequencia de pontos georeferenciados (latitude e longitude), conforme exaustivamente tratado na seção 5.2, podemos nomear cada par de pontos sequenciais desta discretização como P1 e P2, obtendo uma reta r entre estes pontos.

Por outro lado, o registro de dispositivo de GPS do ônibus contêm uma coordenada georeferenciada (latitude e longitude), que seria nosso P, ou seja, o ponto que se deseja projetar ortogonalmente na reta r. No entanto, nem sempre essa projeção é possível, como ilustrado na Figura 37 com dois possíveis pontos P distintos. Na situação ilustrada na direita não seria possível identificar o ponto P', pois ele não existe.

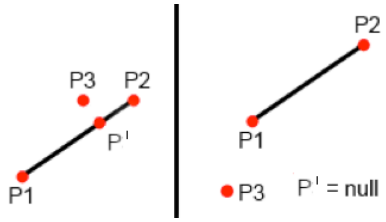


Figura 37 – Projeção ortogonal viável e inviável

Tentando projetar o registro do ônibus em todos os pares de pontos que discretizam o segmento, encontraremos a projeção ortogonal deste registro no segmento ou descobriremos que o registro não se encontra no segmento ao final da testagem de todos os pares. Lembramos que essa testagem foi utilizada em um

dos passos da extração de trajetórias sobre o segmento, conforme detalhado no diagrama apresentado anteriormente.

O pseudo algoritmo implementado para projeção ortogonal é apresentado na Figura 38.

```
projection(p1, p2, p3)
{
   Se(p1.Latitude == p2.Latitude && p1.Longitude == p2.Longitude)
   {
      p1. Latitude = p1.Latitude - 0.00001
      p1.Longitude = p1.Longitude - 0.00001
   }
   var a = ((p3.Latitude - p1.Latitude) * (p2.Latitude - p1.Latitude)) +
            ((p3.Longitude - p1.Longitude) * (p2.Longitude - p1.Longitude))
   var b = Pow(p2.Latitude - p1.Latitude, 2) + Pow(p2.Longitude - p1.Longitude, 2)
   a = a / b
   var p
   p.Latitude = p1.Latitude + (a * (p2.Latitude - p1.Latitude));
   p.Longitude = p1.Longitude + (a * (p2.Longitude - p1.Longitude));
   var minx = Min(p1.Latitude, p2.Latitude;
   var maxx = Max(p1.Latitude, p2.Latitude)
   var miny = Min(p1.Longitude, p2.Longitude)
   var maxy = Max(p1.Longitude, p2.Longitude)
   Se ((p.latitude >= minx && p.latitude <= maxx) &&
       (p.longitude >= miny && p.longitude <= maxy))
      returna p
   else
      returna nulo
}
```

Figura 38 – Pseudo algoritmo projeção ortogonal de p3 na reta p1-p2

O pseudo algoritmo implementado para cálculo de distância é apresentado na **Error! Reference source not found.** e se baseou na fórmula de Haversine [66]. O mesmo considera para o retorna o valor absoluta em metros, com precisão de duas casas decimais, porém é fácil imaginar sua adaptação para outras formas de retorno.

```
distance (p1, p2)
{
   var dLat = (PI / 180) * (p2.Latitude – p1.Latitude)
   var dLon = (PI / 180) * (p2.Longitude - p1.Longitude)
   var a = Sin(dLat / 2) * Sin(dLat / 2) +
           Cos((PI / 180) * (p1.Latitude)) *
           Cos((PI / 180) * (p2.Latitude)) *
           Sin(dLon / 2) * (dLon / 2)
   var c = 2 * Asin(Min(1, Sqrt(a))
   var d = 6371 * c;
   returna Abs(Round(d * 1000, 2))
}
```

Figura 39 – Pseudo algoritmo para cálculo da distância

Retornando a figura que motivou essa explicação de projeção ortogonal e distância, podemos agora utilizar esses conceitos para calcular efetivamente quantos metros foram percorridos pelo ônibus com base nos registros que

descrevem sua trajetória, ou seja, calcular o cumprimento da trajetória. Lembrando, conforme destacado no diagrama anteriormente apresentado, que apenas trajetórias com cumprimento mínimo de 60% do segmento são consideradas.

Percentuais distintos poderiam ser experimentados, mas como o objetivo central era explorar a abrangência da Plataforma BusesInRio e apresentá-la por completa (do dado a entrega do conhecimento), escolhemos um percentual – 60% – que apresentou bons resultados nas experimentações, sem um descarte muito elevado de trajetórias. Em trabalhos futuros experimentações pode ser realizadas para encontrar o percentual adequado desta configuração.

O cálculo deste cumprimento da trajetória corre da seguinte forma:

1) Localizamos o ponto Pi, para o qual o registro de GPS do ônibus do início da trajetória (ponto verde) possui projeção ortogonal Pi' na reta formada por Pi-1 e Pi, onde estes são pontos sequenciais que discritizam o segmento (e.g. cada coordenada em sequencia do campo Coordenadas da Tabela 17). Na existência de mais de um segmento monitorado o procedimento se repete para cada um deles.

2) O mesmo é feito para encontrar o ponto Pf, para o qual o registro de GPS do ônibus do fim da trajetória (ponto vermelho) possui projeção ortogonal Pf' na reta formada por Pf-1 e Pf, onde estes são pontos sequenciais que descritizam o segmento.

3) Como Pi e Pf são pontos que discretizam o segmento, podemos calcular a distância entre eles somando a distância entre cada par de pontos sequencias entre os mesmos, incluindo os próprios. Essa distância chamamos de ds.

4) Precisamos considerar que d não inclui a distância de Pi para Pi' e de Pf' para Pf, ou seja, as distâncias di = distance(Pi,Pi') e df = distance(Pf',Pf).

5) Dessa forma, a distância percorrida pela trajetória pode ser obtida calculando-se d = ds - df + di. Repare que as projeções são sempre antes de Pi e Pf, o que implica que deixamos de considerar distância no início e consideramos distâncias a mais no fim, por isso a diferença de sinal na fórmula apresentada. A Figura 40 ilustra esses raciocínios anteriormente descritos.

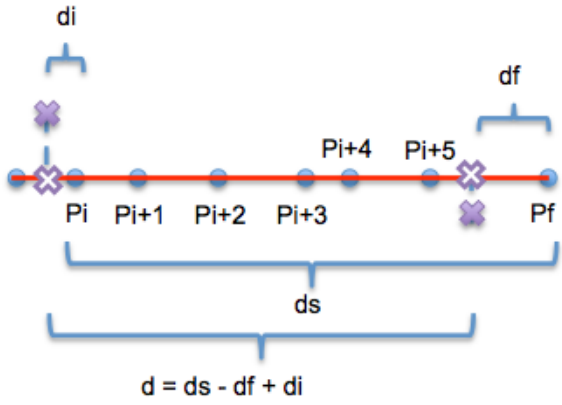


Figura 40 – Cálculo da distância do segmento

Isto posto, somos agora capazes de calcular a distância percorrida pelo ônibus em uma passagem pelo segmento com base nos dados de seus dispositivos de GPS. Da mesma forma, somos capazes de calcular o tempo necessário para percorrer tal distância, utilizando o dado temporal de coleta nos dispositivos. Consequentemente, podemos estimar a velocidade média da trajetória. De igual forma, podemos analisar individualmente pares de dados destas trajetórias.

No software algumas informações adicionais são apresentadas ao passar o mouse sobre um registro da trajetória (veja Figura 41).

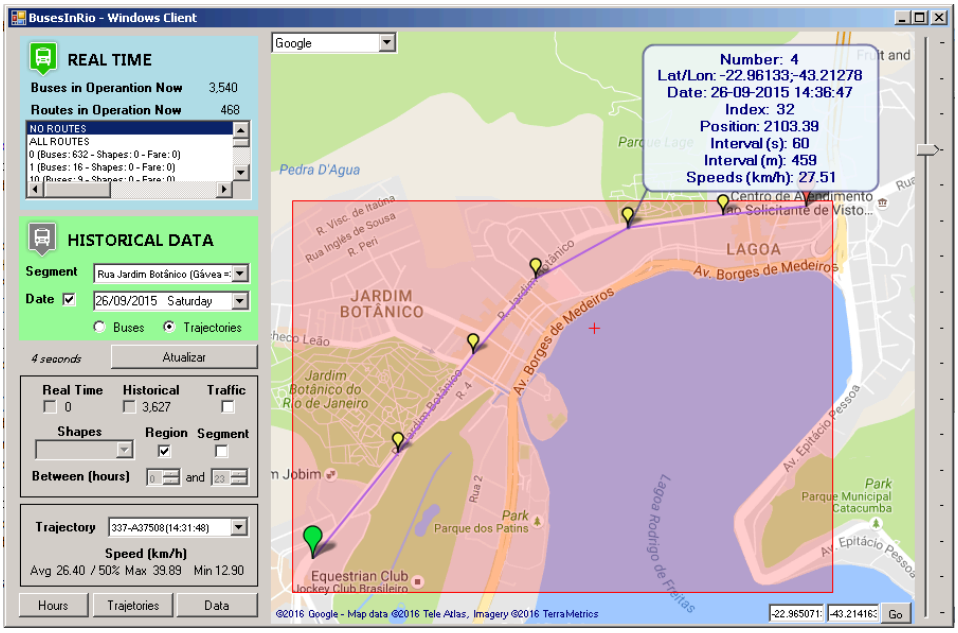


Figura 41 – Informações adicionais da trajetória no WinDemo

Esses informações são respectivamente: “number” a posição da sequencia de registros que descreve a trajetória, sendo o primeiro o zero; “Lat/Lon” as

coordenadas do registro; “Date” o momento da coleta do registro no dispositivo GPS do ônibus; “index” o índice do ponto que discritiza o segmento (i.e. qual a posição das Coordenadas da Tabela 17), seguinte a projeção ortogonal destas coordenadas no segmento em questão (trata-se do “x” do Px ilustrado em Figura 40); “position” o valor percorrido do segmento em metros; “interval (s)“ o intervalo temporal em relação ao registro anterior; “interval (m)” o intervalo em distância em relação ao registro anterior; “speed (km/h)” a velocidade média entre o registro e seu anterior em quilômetros por hora.

No software pode também ser obtido um gráfico plotando a evolução das trajetórias da relação da sua duração em segundos e do seu comprimento em metros, ao se clicar no botão “Trajectories”, conforme ilustrado na Figura 42. Este gráfico permite identificar alguma trajetória discrepante para uma análise individualizada.

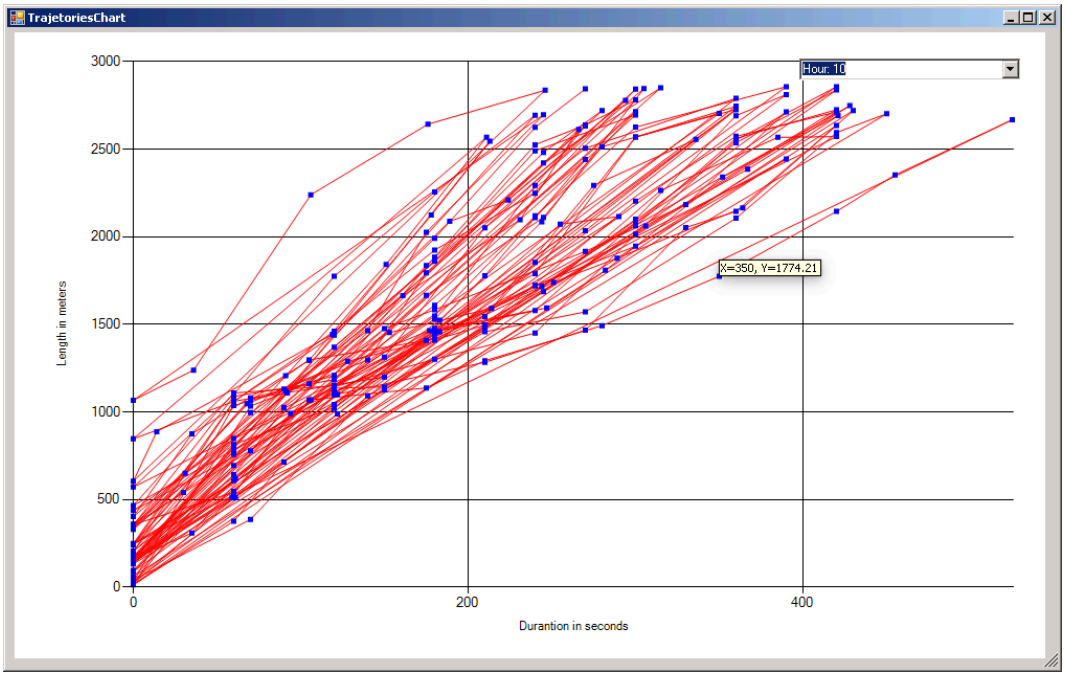
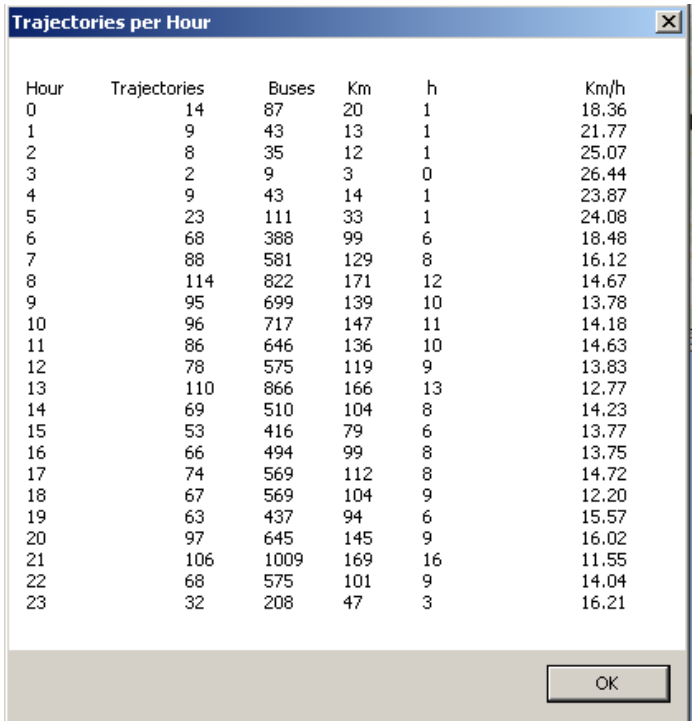


Figura 42 – Gráfico duração x percurso da trajetória no WinDemo

Observe que os mesmos filtros por horário de início e de trajetória individualmente estão também disponíveis no gráfico. Além disso, é possível passar o mouse sobre um dos registros (pontos azuis) para visualizar os valores exatos, conforme ilustrado. Cada linha vermelha representa uma trajetória.

5.4
Análise Descritiva das Trajetórias

Considerando que cada trajetória tem sua velocidade média, podemos agrupá-las pelo horário de início para obter a velocidade média de cada hora exata. Essa média permite entender o comportamento do trafego no segmento ao longo do dia, conforme a Figura 43.



Hour	Trajectories	Buses	Km	h	Km/h
0	14	87	20	1	18.36
1	9	43	13	1	21.77
2	8	35	12	1	25.07
3	2	9	3	0	26.44
4	9	43	14	1	23.87
5	23	111	33	1	24.08
6	68	388	99	6	18.48
7	88	581	129	8	16.12
8	114	822	171	12	14.67
9	95	699	139	10	13.78
10	96	717	147	11	14.18
11	86	646	136	10	14.63
12	78	575	119	9	13.83
13	110	866	166	13	12.77
14	69	510	104	8	14.23
15	53	416	79	6	13.77
16	66	494	99	8	13.75
17	74	569	112	8	14.72
18	67	569	104	9	12.20
19	63	437	94	6	15.57
20	97	645	145	9	16.02
21	106	1009	169	16	11.55
22	68	575	101	9	14.04
23	32	208	47	3	16.21

Figura 43 – Exemplo da tela de trajetórias por horário do WinDemo

A primeira coluna (“Hour”) apresenta a hora para o qual se realizou a consolidação, ou seja, foram utilizados dados de todos os trajetos que se iniciaram naquela hora do dia, do seu primeiro ao último segundo. A segunda coluna (“Trajectories”) totaliza o número de trajetórias que foram utilizadas para consolidação. Naturalmente, em horários de maior fluxo temos mais ônibus na rua, logo um número maior de trajetórias, ou melhor, sensores. Na sequencia temos o número de registros de GPS dos Ônibus contidos nestas trajetórias (“Buses”). As colunas seguintes apresentam o total de quilômetros (“Km”) e horas (“h”) destas trajetórias. A última coluna (“Km/h”) apresenta a média das velocidades médias. Salientamos que não é a divisão do total de quilômetros percorridos pelo tempo em horas, mas a média das velocidades médias das trajetórias. Em trabalhos futuros, podem ser experimentadas diferentes abordagens estatísticas para cálculo da melhor descrição do comportamento do tráfego em

determinada hora, no entanto, optamos pela abordagem supracitada para facilitar a compreensão do leitor, uma vez que o objetivo central é demonstrar de forma completa e abrangente o uso da Plataforma BusesInRio.

Considerado a maior e menor média do dia como referências, podemos adotar uma escala de cores para avaliar o fluxo em cada horário, conforme a Figura 44.

Label	Color	Relative speed (%)
Light		[100, 60]
Moderate		[50, 40]
Heavy		[30, 20]
Very Heavy		[10, 0]

Figura 44 – Escala de cores para os segmentos em função do tráfego

O software em questão aplica essa escala e apresenta a velocidade máxima, mínima e a média do horário no canto esquerdo inferior de sua interface, conforme ilustrado, quando a opção “Traffic” é selecionada junto a uma hora nas trajetórias, conforme Figura 45. A Figura 43 pode ser obtida pelo botão “Hours”.

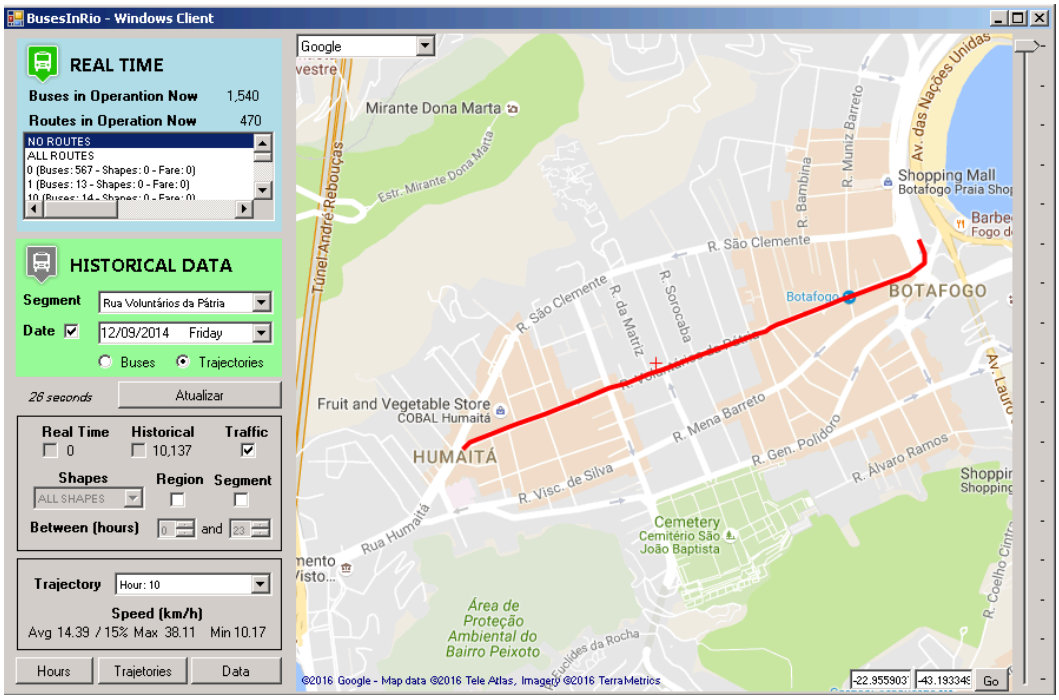


Figura 45 – Escala de cores para o segmento às 10 horas

A abordagem empregada pode ser expandida da visão de apenas um dia para a comparação entre dias e diferentes horários destes, permitindo assim, a identificação de padrões e a comparação entre diferentes situações.

5.5
Identificação de Padrões para os Segmentos

Conforme demonstrado ao longo do capítulo, a Plataforma BusesInRio oferece um método para acessar as trajetórias dos ônibus realizadas sobre um segmento conhecido em determinada data (i.e. método `GetTrajetoriesByDateAndSegment`), conhecimento o qual é extraído pelas instâncias do serviço *Extraction* da Plataforma. A partir desses dados é possível consolidar estatísticas descritivas do tráfego do dia, como inicialmente apresentado na seção 5.3. A fim de evitar um fluxo desnecessário de dados, apenas esses dados consolidados podem ser consumidos pelo método chamado TrafficSummaryDay.

A seguir, para alguns segmentos escolhidos, apresentamos a evolução da velocidade média de cada dia ao longo do tempo junto a um indicador do número de trajetórias, em milhares, utilizadas para este cálculo. Os segmentos analisados são respectivamente: Rua Jardim Botânico (Humaitá => Gávea), Rua São Clemente, Rua Voluntários da Pátria, Av. Ns. Sra. de Copacabana e Rua Barata Ribeiro.

Uma conclusão interessante que podemos chegar ao visualizar esses gráficos é que existe uma tendência na redução do número de ônibus passando pelos segmentos analisados, percebido de forma mais acentuada na Av. Ns. Sra. de Copacabana, ou seja, cronologicamente o número de ônibus passando pelos segmentos analisados está diminuindo. Ao mesmo tempo, percebesse uma leve redução na velocidade média na maioria dos segmentos, excetuando-se os dois últimos. Tal comportamento pode se justificar pelo esforço da Prefeitura em reduzir o número de ônibus trafegando pela Zona Sul da Cidade, onde esses segmentos estão inseridos.

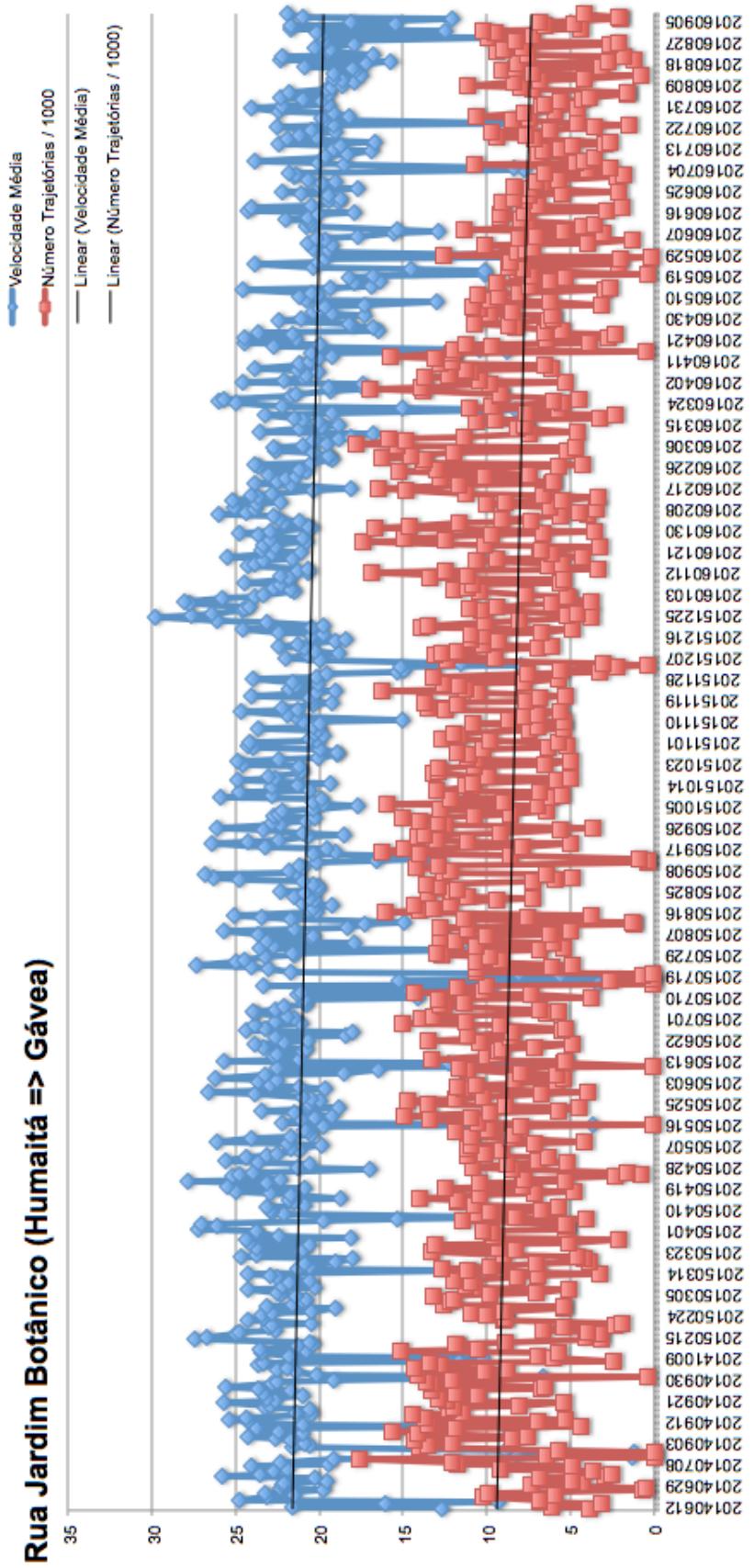


Figura 46 – Velocidade Rua Jardim Botânico (Humaitá => Gávea)

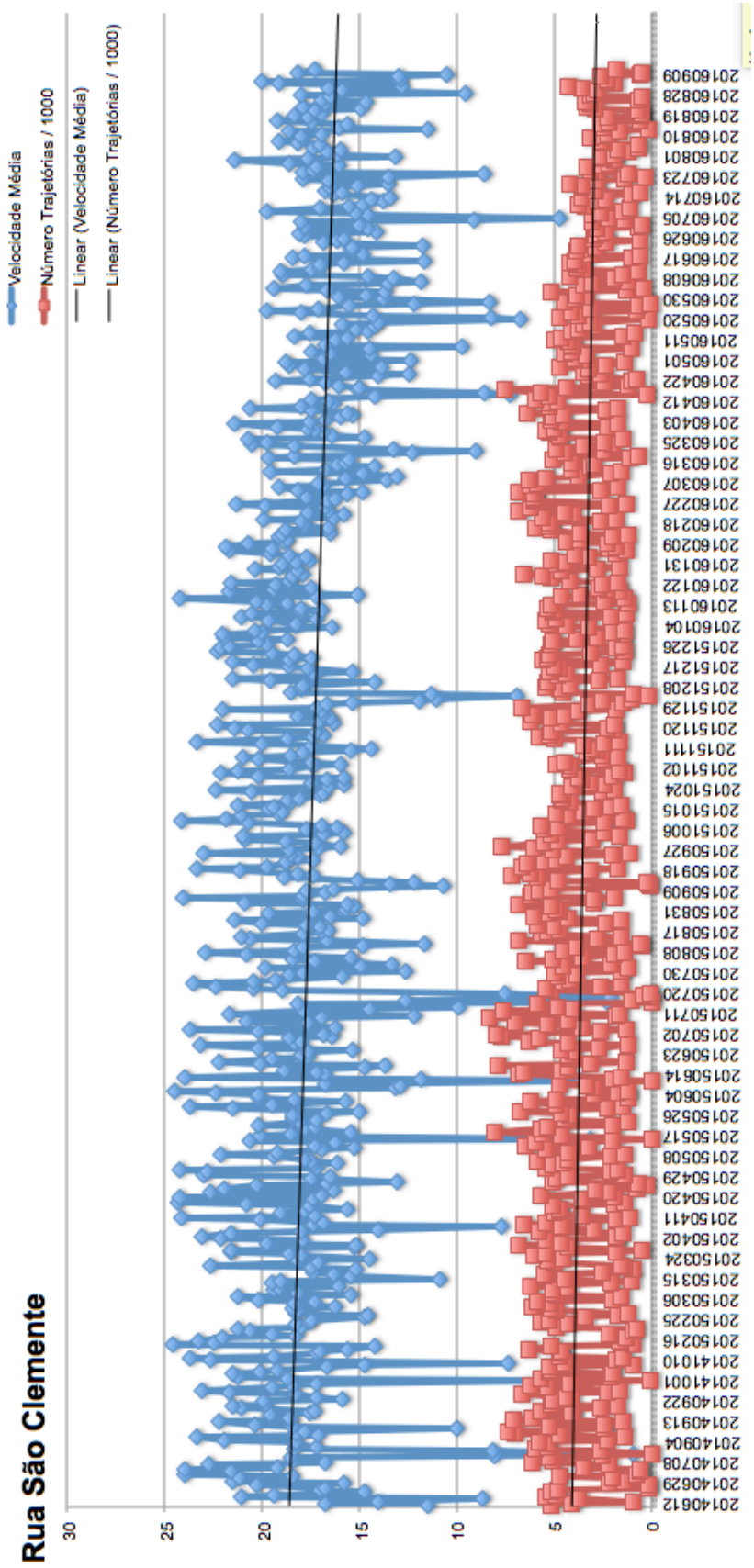


Figura 47 – Velocidade Rua São Clemente

Figura 48 – Velocidade Rua Voluntários da Pátria

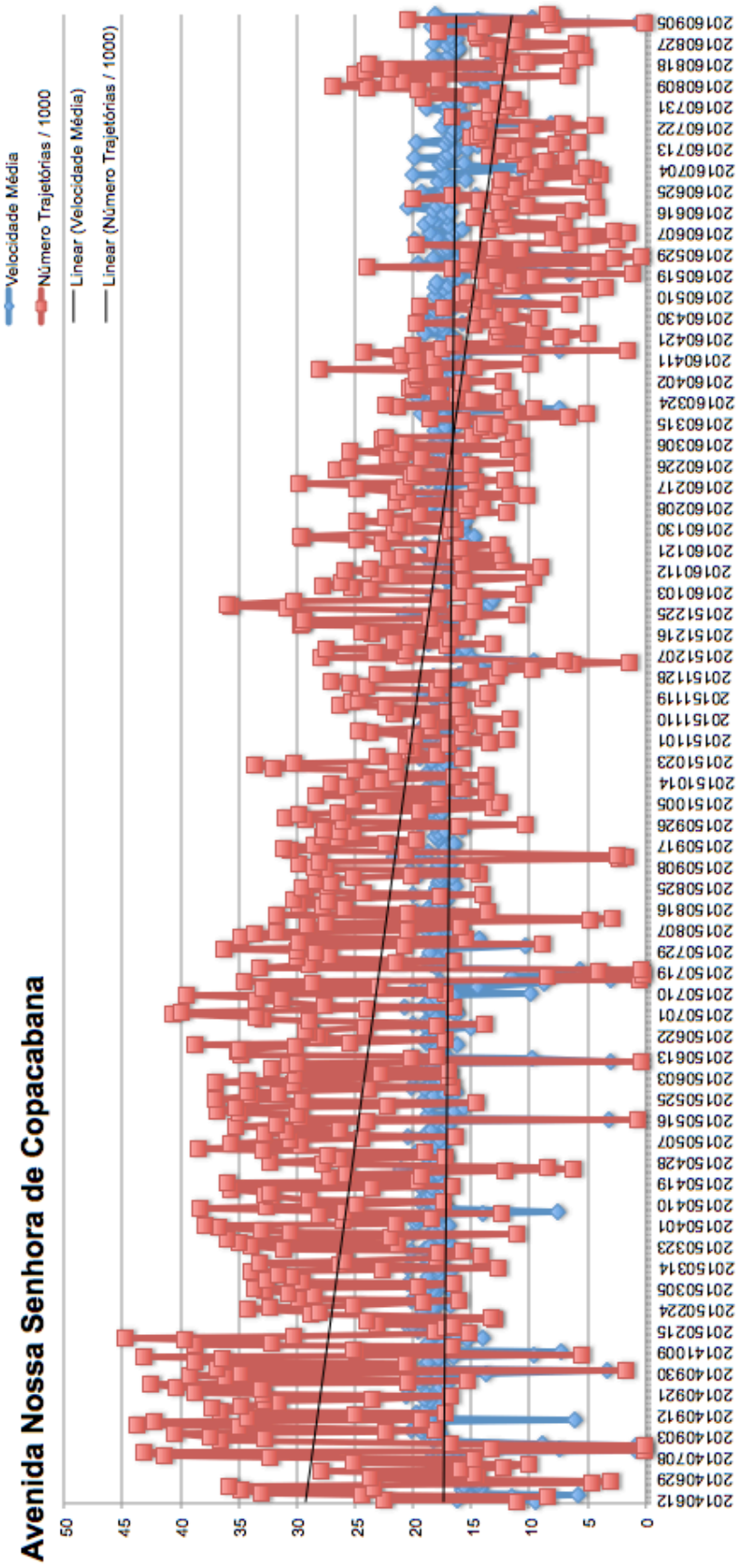


Figura 49 – Velocidade Av. Ns. Sra. de Copacabana

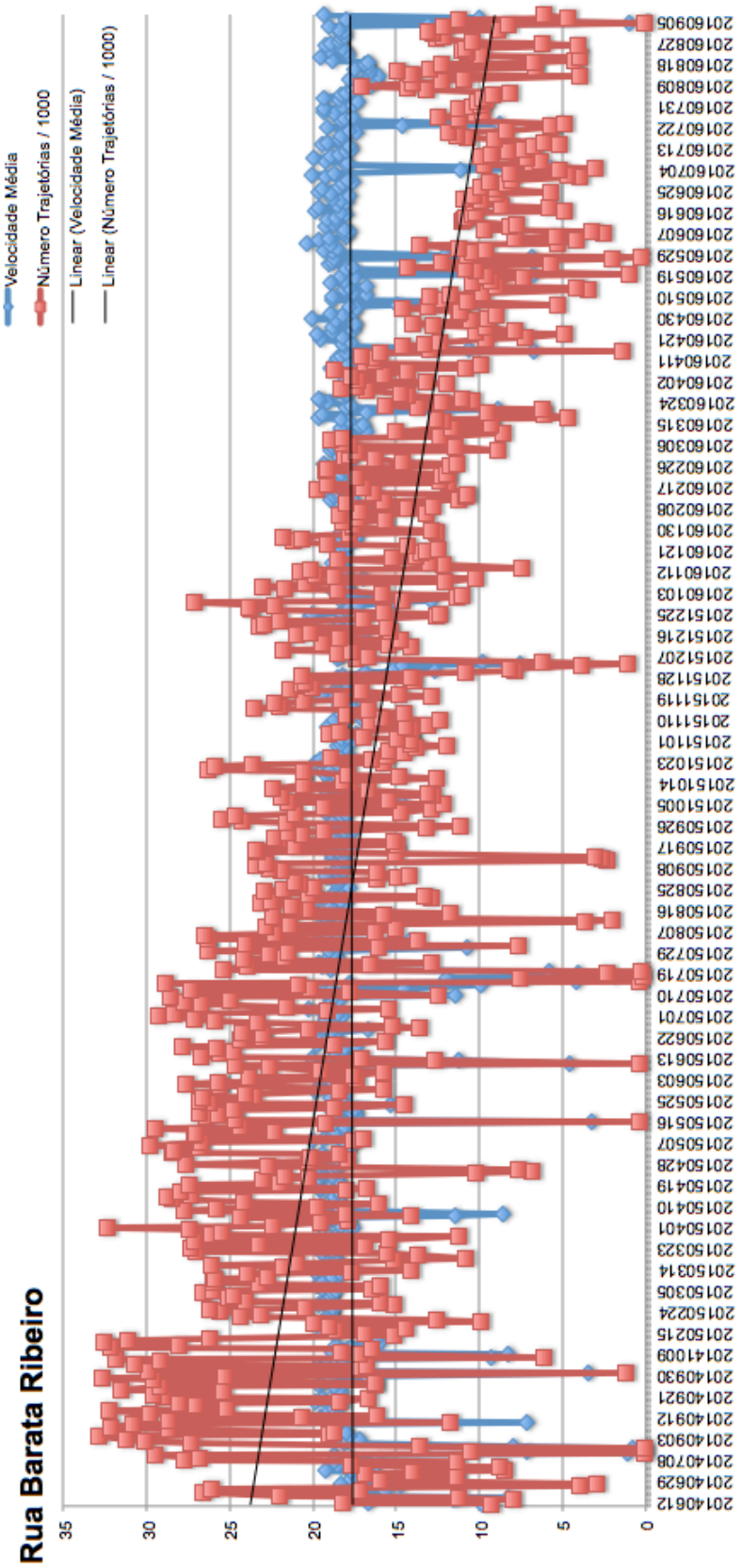
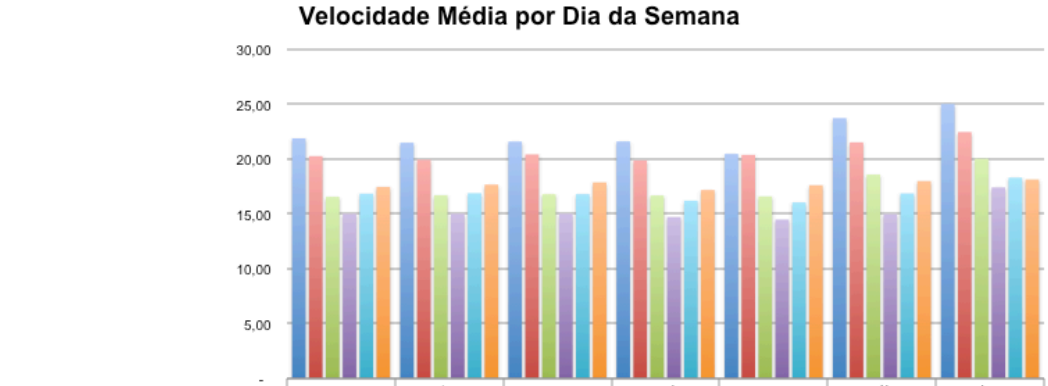


Figura 50 – Velocidade Rua Barata Ribeiro

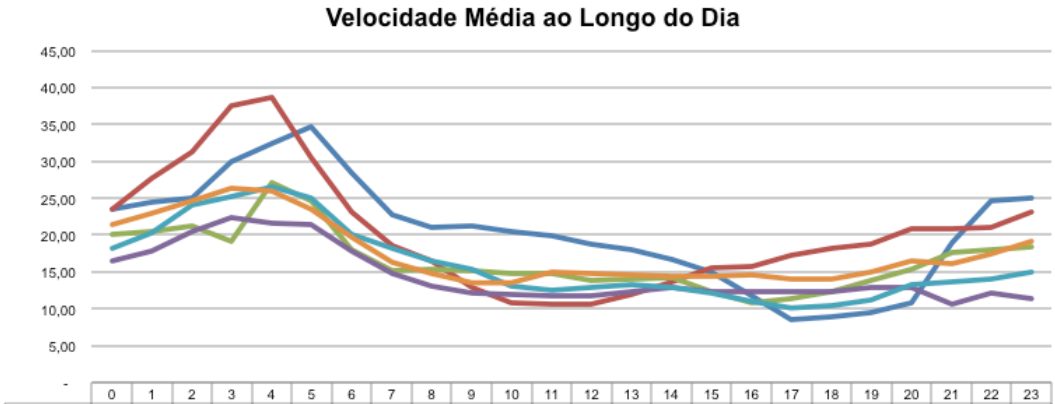
Outra visão interessante é o comportamento do transito nos diferentes dias da semana nestes mesmos segmentos anteriormente citados. Como esperado, final de semana a velocidade média em todos os segmentos é superior em relação aos demais dias, conforme demonstrado no Figura 51.



	seg	ter	qua	qui	sex	sáb	dom
Rua Jardim Botânico (Humaitá => Gávea)	21,87	21,47	21,57	21,59	20,45	23,71	24,99
Rua Jardim Botânico (Gávea => Humaitá)	20,25	19,91	20,41	19,86	20,36	21,50	22,44
Rua São Clemente	16,54	16,70	16,78	16,66	16,58	18,56	20,02
Rua Voluntários da Pátria	14,98	15,00	14,99	14,71	14,47	14,97	17,40
Av. Ns. Sra. de Copacana	16,83	16,88	16,78	16,19	16,03	16,86	18,29
Rua Barata Ribeiro	17,45	17,65	17,86	17,16	17,60	17,98	18,12

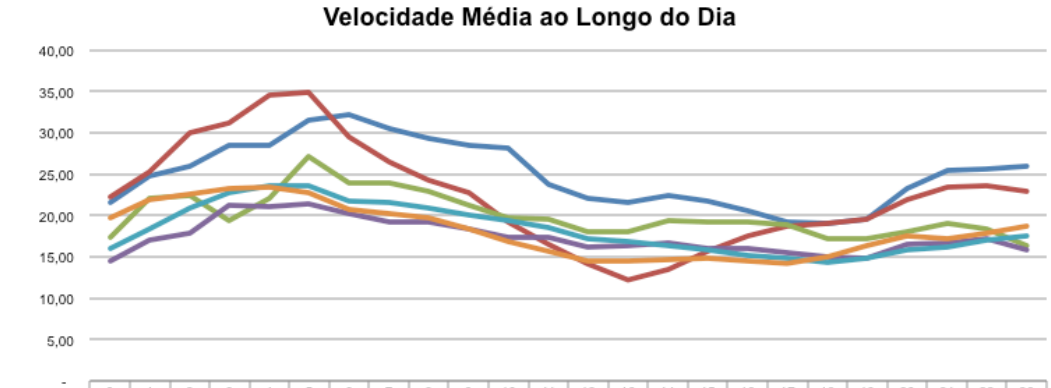
Figura 51 – Velocidade ao Longo dos Dias da Semana

A Figura 52 apresenta a variação da velocidade média histórica ao longo das faixas de horário de um dia, em particular, da média das sextas-feiras, adotando a mesma ordem para os segmentos da figura anterior. Em contraste, o Figura 53 apresenta a variação ao longo dos domingos.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Series1	23,41	24,54	25,11	30,00	32,45	34,69	28,44	22,84	21,06	21,20	20,43	19,99	18,68	17,97	16,66	14,99	11,76	8,52	9,00	9,39	10,88	19,01	24,71	24,97
Series2	23,48	27,65	31,22	37,51	38,66	30,47	23,10	18,50	16,43	12,95	10,87	10,58	10,68	11,86	13,60	15,57	15,77	17,27	18,14	18,78	20,76	20,81	21,05	23,06
Series3	20,18	20,39	21,31	19,16	27,03	24,62	17,97	15,12	15,29	15,20	14,85	14,74	13,87	14,07	14,26	12,28	10,80	11,33	12,40	13,80	15,40	17,56	17,97	18,42
Series4	16,49	17,82	20,39	22,39	21,56	21,36	17,82	14,85	13,05	12,17	11,94	11,83	11,68	12,24	12,81	12,38	12,26	12,29	12,31	12,85	12,89	10,62	12,06	11,35
Series5	18,22	20,28	24,14	25,29	26,56	25,02	20,04	18,16	16,42	15,32	13,12	12,53	12,79	13,35	12,95	12,15	11,06	10,11	10,39	11,13	13,21	13,63	13,95	14,94
Series6	21,36	22,92	24,64	26,35	26,06	23,52	19,71	16,30	14,70	13,52	13,52	14,91	14,77	14,62	14,33	14,34	14,53	14,03	14,03	15,06	16,50	16,17	17,35	19,17

Figura 52 – Velocidade ao Longo das Horas das Sextas-feiras



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Series1	21,52	24,69	25,99	28,53	28,46	31,49	32,14	30,57	29,39	28,45	28,21	23,71	22,00	21,57	22,38	21,72	20,62	19,19	19,02	19,53	23,33	25,46	25,69	26,00
Series2	22,18	25,21	29,94	31,25	34,48	34,92	29,56	26,46	24,23	22,68	19,28	16,47	14,07	12,20	13,52	15,66	17,52	18,72	19,08	19,48	21,87	23,43	23,55	22,85
Series3	17,43	22,08	22,46	19,37	22,03	27,11	23,92	23,88	22,86	21,29	19,75	19,61	17,96	18,03	19,29	19,13	19,23	18,87	17,17	17,21	18,07	18,97	18,40	16,26
Series4	14,42	17,03	17,82	21,30	21,11	21,45	20,26	19,15	19,13	18,36	17,36	17,37	16,16	16,33	16,70	16,02	16,07	15,55	15,04	14,89	16,49	16,62	17,15	15,86
Series5	15,97	18,44	20,96	22,68	23,52	23,63	21,78	21,51	20,96	20,12	19,32	18,54	17,13	16,79	16,29	15,79	15,18	14,84	14,30	14,74	15,81	16,22	16,96	17,44
Series6	19,77	21,82	22,65	23,23	23,40	22,73	20,79	20,21	19,75	18,42	16,79	15,65	14,41	14,48	14,72	14,75	14,54	14,20	14,91	16,40	17,49	17,22	17,86	18,69

Figura 53 – Velocidade ao Longo das Horas dos Domingos

As figuras apresentadas, obviamente poderiam subsidiar complexas análises estatísticas de cada um destes segmentos e de qualquer outro com sensores suficientes (questão tratada na seção 5.2), estando essas trajetórias, suas estatísticas descritivas e as inúmeras possíveis perspectivas de consolidação franqueadas aos desenvolvedores que queiram consumir estes métodos da Plataforma BusesInRio, por exemplo, para desenvolver ferramentas de análises de tráfego com propósitos específicos. Nesta seção nos limitamos a consolidar essas estatísticas utilizando planilhas para ilustrar o poder do conhecimento extraído e, também, que o padrão do transito nestes segmentos varia ao longo do tempo (anos, meses, dia, horas, etc.). Desta forma, a escolha de que datas considerar para composição do padrão é extremamente relevante, assim como, a manutenção da subdivisão dos dados por faixa de horários para uma melhor sensibilidade.

Diante desta necessidade de selecionar datas específicas, a Plataforma BusesInRio oferece o método TrafficSummaryPeriod para retornar apenas a média das datas desejadas, mantendo a visão dividida pelo horário e a total descriminadas. Em todos os métodos mencionados nesta seção, o conhecimento base é o mesmo que detalhado na seção 5.4, as trajetórias sobre os segmentos. O que varia é o período para o qual é feito a média (das trajetórias do dia ou da média das trajetórias dos dias selecionados). A precisão utilizada é sempre de duas casas decimais e o resultado comtempla sempre o total e o detalhamento por faixa de hora.

Analisar o padrão do tráfego no segmento torna-se então um simples exercício de escolher datas e consumir esse método. No entanto, conforme ficou claro com as figuras apresentadas nesta seção, essa escolha influencia significativamente na velocidade média esperada como padrão para o segmento e suas faixas de horário. É intuitivo imaginar que o tráfego apresente comportamentos distintos em situações diferentes, como, por exemplo:

- período de férias x período letivo;
- dias úteis, x fins de semana x feriados;
- período com um evento nas proximidades x sem impactos na região;
- acidente no segmento ou nos segmentos seguintes a este x sem acidente;
- intervenção nas vias do segmento x via normal;
- alterações nas vias do segmento x via sem alteração; e
- dentre inúmeras outras possibilidades.

Por outro lado, é importante acrescentarmos na discussão o ponto de vista do usuário deste padrão de tráfego, pois se estivermos falando, por exemplo, de um profissional que consume essa informação interessado em analisar o impacto positivo ou negativo de alterações nos semáforos ao longo do tempo, todas essas variáveis são relevantes para uma adequada comparação entre dados recentes e históricos, mas se estivermos falando de um cidadão, morador da área urbana, que está interessado em saber se o transito está bom ou ruim, o seu referencial é mais imediato, ou seja, tem maior relação com sua memória recente sobre o comportamento da via.

Nesse sentido, propormos combinar uma visão histórica com uma visão do comportamento recente para obter a velocidade padrão (VMP) – esperada – em um horário, data e segmento em particular. Portanto, devemos considerar a média de: (a) todos os dados históricos (VMH); (b) todos os dados históricos para o mesmo dia da semana (VMD); (c) dados apenas dos cinco dias de igual dia da semana que o antecederam (VM5D); e (d) dados dos últimos sete dias anteriores (VM7H).

Tomando o dia 31/08/2016 como de interesse, ou seja, para o qual desejamos calcular o padrão esperado para um determinado segmento, no caso, o segmento “Rua São Clemente”, teríamos o VMP da Tabela 18, com seus

respectivos VMH, VMHD, VM5D e VM7H utilizados na sua média. Nesta tabela foi também incluído a velocidade média dos horários deste dia, posteriormente extraídas, para que se avalie o decorrer deste dia em relação ao padrão calculado, o que é sintetizado na última coluna com a divisão deste valor extraído por VMP.

Horário	VMH	VMHD	VM5D	VM7H	VMP	31/08/16	Dia/VMP
0	19,46	21,04	23,95	21,57	21,51	0	N/A
1	21,26	21,45	18,622	12,43	18,44	29,82	162%
2	22,21	21,92	21,134	20,54	21,45	25,42	119%
3	21,00	23,09	30,546	10,28	21,23	24,92	117%
4	25,62	26,45	25,872	17,16	23,77	30,86	130%
5	24,16	23,73	23,356	19,66	22,73	20,45	90%
6	18,68	17,36	17,504	15,35	17,22	15,23	88%
7	16,15	13,92	14,816	15,40	15,07	12,39	82%
8	16,22	14,48	14,568	14,98	15,06	14,12	94%
9	15,97	14,71	13,902	15,92	15,12	13,88	92%
10	15,58	14,47	14,614	14,72	14,84	13,98	94%
11	15,62	14,79	14,526	15,16	15,02	13,62	91%
12	14,52	13,34	13,432	13,83	13,78	10,68	77%
13	14,84	13,72	14,706	14,74	14,50	12,46	86%
14	15,07	13,92	14,42	14,02	14,36	12,64	88%
15	14,28	12,36	13,592	13,71	13,48	11,11	82%
16	13,69	12,03	13,684	14,19	13,40	8,57	64%
17	13,33	11,79	14,088	13,90	13,28	7,94	60%
18	13,13	11,62	12,344	14,32	12,86	12,58	98%
19	14,63	13,79	15,122	15,65	14,80	14,37	97%
20	16,31	15,44	16,21	16,99	16,24	16,62	102%
21	18,00	17,29	18,122	18,08	17,88	15,86	89%
22	18,54	19,05	18,512	18,47	18,64	19,55	105%
23	18,95	20,99	18,79	16,59	18,83	24,5	130%

Tabela 18 – Cálculo Velocidade Média Padrão para 31/08/2016

Lembramos que a velocidade média de um determinado horário, data e segmento foi calculada através das trajetórias iniciadas neste horário, conforme amplamente debatido nas seções 5.3 e 5.4. Além disso, cabe registrar que nas médias valores zerados ou inexistente são desconsiderados, dado que essas situações indicam que não existem trajetórias válidas suficientes sobre o segmento naquela data e horário ou mesmo não existem dados deste momento armazenados, sendo as médias mencionadas realizadas sem incluir este valor, mas não o substituindo por um anterior, em todos os casos supracitados.

Em relação a Tabela 18, VMH utilizou todas as velocidades médias desde 12/06/2014 até 30/08/2016 para calcular a velocidade média para cada horário deste segmento. Enquanto VMHD utilizou o mesmo período, mas apenas datas que caíram em quartas-feiras, dado que 31/08/2016 foi uma quarta-feira. Raciocínio análogo ocorreu para selecionar as datas de 24/08, 17/08, 10/08, 03/08 e 27/07/2016, todas quartas-feiras, para calcular VM5D. Por último, apenas as datas de 30, 29, 28, 27, 26, 25 e 24/08/2016 foram consideradas para calcular VM7H.

Repare que a última coluna da Tabela 18 permite sugestionar que o dia em questão, 31/08/2016, no segmento "Rua São Clemente", foi dentro do esperado, na maioria dos horário, se considerado que até 80% do padrão encontra-se dentro da normalidade, excetuando-se as faixas de 16 e 17h, que ficaram na ordem de 60%.

Frente ao exposto, a Plataforma BusesInRio pode manter calculada a velocidade media padrão para os horários e segmentos monitorados. No início de cada dia, esses valores podem ser atualizados, em função dos dados capturados no dia anterior.

Salientamos que a estratégia de cálculo adotada não é definitiva, uma vez que analises estatísticas mais profundas podem ser realizadas e até sutilezas específicas podem ser consideradas, sendo assim implementadas formulas de cálculo mais assertivas do pronto de vista preditivo. No entanto, a presente seção cumpre com o objetivo de ilustrar o conhecimento que a Plataforma BusesInRio tem a sua disposição para apoiar essas decisões.

5.6
Monitoramento do Tráfego em Segmentos

O algoritmo apresentado anteriormente para extração de trajetórias sobre segmentos é todo baseado no recebimento de registros sequencias dos dispositivos dos Ônibus. Portanto, esse pode ser aplicado para processamento de dados históricos, enviados a partir da leitura de dados armazenados de forma sequencial, como exaustivamente comentado anteriormente, assim como, também pode ser usado a partir dos últimos dados capturados, ou seja, pode monitorar o tráfego em tempo "quase" real.

Ressaltamos o cuidado em destacar que existe um atraso entre a situação real e a identificação do tráfego, oriundo do atraso no recebimento dos dados, de minuto a minuto, mas também pela necessidade de aguardar o ônibus finalizar seu trajeto sobre o segmento para podermos extrair esse conhecimento e avaliar os indicadores da trajetória.

Outra distinção importante é que na lógica histórica aguardávamos todos os trajetos realizados em um segmento, iniciados em determinado horário, serem

extraídos, para consolidarmos indicadores, portanto, particularidades de uma trajetória eram diluídas para o horário. Além disso, trajetórias que não atendiam minimamente a alguns critérios eram descartadas.

Diante dos argumentos expostos, julgamos adequado considerar um número mínimo de trajetórias no segmento em determinado período para extrair um indicador. Sendo assim, propomos considerar as últimas 3 trajetórias satisfatórias para determinar a situação atual do transito, considerando ainda, que além das checagens de trajetória já adotadas, não escolheremos trajetórias com intervalo de conclusão superior a 30 minutos entre elas. Os critérios para validação de uma trajetória foram apresentados anteriormente (veja Figura 31).

O método `TrajectoriesNow` da Plataforma BusesInRio fornece as trajetórias que atendem essas características para um segmento em específico, nos moldes das trajetórias extraídas do histórico. Assim como, o método `TrafficNow` da Plataforma fornece a velocidade média destas trajetórias. Esse método retorna também a velocidade padrão esperada para aquele momento, conforme debatido na seção anterior. Assim, essas velocidades podem ser comparadas (situação real x padrão) para definição de uma escala de cores ou outras indicações. No próximo capítulo abordaremos formas de explorar esses conhecimentos visando o usuário final.

5.7 Análise Cruzada de Segmentos

Uma perspectiva interessante de ser analisada é a relação entre segmentos que se sucedem de forma direta ou indireta, ou seja, ruas conectadas diretamente umas as outras no mesmo sentido de tráfego ou que sejam conectadas por uma rua comum entre as mesmas, preservando o sentido entre as três. Por exemplo, o segmento “Rua São Clemente” tem como um dos possíveis destinos o segmento “Rua Jardim Botânico (Humaitá => Gávea)”, no caso, sendo estes conectados pela “Rua Humaitá”, todos no mesmo sentido. Digamos então que estes podem ser apelidados respectivamente de A, C e B, onde de A é possível ir para B e de B é possível ir para C. Desta forma, é intuitivo que uma piora no tráfego em C possa refletir em uma piora A, uma vez que parte dos veículos deste segundo segmento estarão indo para o primeiro.

A Tabela 18 apresentou a velocidade padrão a ser esperada no segmento C em 31/08/2016, assim como, a velocidade que ocorreu por faixa de horário nesta data. A mesma metodologia utilizada para construção desta tabela permitiu elaborar a Tabela 19 para a mesma data em relação a A.

Horário	VMH	VMHD	VM5D	VM7H	VMP	31/08/16	Dia/VMP
0	23,62	24,69	21,846	25,23	23,85	30,91	130%
1	27,11	28,71	26,886	12,43	23,79	32,32	136%
2	29,49	31,80	33,822	20,54	28,91	37,18	129%
3	31,71	33,88	29,93	10,28	26,45	33,46	127%
4	32,79	36,69	41,276	17,16	31,98	37,17	116%
5	33,60	35,50	35,812	19,66	31,14	38,03	122%
6	29,09	28,19	28,494	15,35	25,28	28,68	113%
7	23,81	21,30	23,712	15,40	21,06	19,63	93%
8	21,55	18,87	20,796	14,98	19,05	20,55	108%
9	21,75	19,65	20,738	15,92	19,51	22,66	116%
10	21,71	19,56	18,978	14,72	18,74	22,62	121%
11	20,89	19,08	17,416	15,16	18,14	21,68	120%
12	19,61	17,84	17,292	13,83	17,14	21,41	125%
13	19,22	17,95	17,866	14,74	17,44	21,21	122%
14	19,13	18,07	16,262	14,02	16,87	21,11	125%
15	18,56	17,74	16,762	13,71	16,69	17,1	102%
16	15,16	13,36	12,406	14,19	13,78	8,99	65%
17	10,75	7,38	6,458	13,90	9,62	6,39	66%
18	11,19	7,94	6,82	14,32	10,07	5,56	55%
19	12,14	8,63	7,678	15,65	11,02	5,68	52%
20	14,60	10,80	10,788	16,99	13,30	7,04	53%
21	22,45	22,19	22,026	18,08	21,19	24,67	116%
22	26,14	27,79	25,644	18,47	24,51	25,31	103%
23	26,97	30,09	27,236	16,59	25,22	27,48	109%

Tabela 19 – Cálculo Velocidade Média Padrão para 31/08/2016 em A

Se focarmos a atenção apenas para variações negativas de mais de 20% entre a velocidade padrão e a constatada, temos um momento especial a ser analisado, que separamos na Tabela 20, para os segmentos A e C respectivamente.

Horário	A	C
16	65%	64%
17	66%	60%
18	55%	98%
19	52%	97%
20	53%	102%

Tabela 20 – Destaque comparação A x C

De fato, a análise deste destaque demonstra uma lentidão perceptível em C durante as faixas de horário de 16 e 17h, na casa de 40% negativa em relação ao padrão de cada horário, que provavelmente influenciou em uma lentidão em A, que se propagou da faixa de 16 até a de 20 horas, nesta mesma ordem.

Diante do exemplo apresentado, registra-se que os conhecimentos extraídos para cada segmento podem ser combinados de forma a um conhecimento ainda mais abrangente ser alcançado, permitindo assim, um entendimento da cidade como um todo. Inclusive, estes podem servir para abordagens focada na identificação de anomalias no tráfego e suas consequências nos segmentos anteriores. No entanto, não é objetivo deste trabalho aprofundar a análise de anomalias e sua propagação pela cidade, mas este é um tópico, sem dúvida, muito interessante, que pode ser posteriormente desenvolvido se utilizando a Plataforma BusesInRio proposta neste trabalho.

5.8
Correção do Erro de GPS e Cálculo de Comprimento

Cabe registro, antes de evoluirmos para outros algoritmos, alguns conhecimentos que podem ser extraídos a partir de técnicas detalhadas anteriormente.

- **Descarte de Registro:** Os dispositivos de GPS apresentam uma margem de erro em geral na ordem de 100 metros. Portanto, extrapolando, podemos adotar a definição de que um ponto distante em mais de 200 metros de uma coordenada válida não pode ser considerado correto. Entendendo coordenada válida, por exemplo, como pontos que discretizam um segmento ou uma linha de ônibus, dado que essas foram visualmente validados frente aos mapas da cidade. Como o cálculo de distância foi apresentada anteriormente, não se faz necessário detalhamento adicional deste raciocínio.
- **Correção do Erro de GPS:** Os pontos não descartados, cuja discretização da linha de ônibus correspondente é conhecida, podem ser corrigidos, ou seja, a localização do ônibus pode ser projetado exatamente para interpolação destes pontos que a discretizam. Se existe apenas uma projeção válida a projeção pode ser adotada para exibição da posição em questão do ônibus. Senão, cabe escolher a mais próxima do ponto com erro. Como o cálculo de projeção ortogonal foi apresentada anteriormente, não se faz necessário detalhamento

adicional deste raciocínio. Esse conhecimento está disponível no método `GetBusDetailsWithProjection` da Plataforma.

- **Cálculo de Comprimento Registro:** Conforme explorado anteriormente, o comprimento do segmento pode ser calculado somando a distância entre os pares de pontos sequenciais que discretizam a mesma. Raciocínio idêntico pode ser adotado para calcular para os trajetos de linha de ônibus. Esses conhecimentos estão respectivamente disponíveis nos métodos `GetSegmentLength` e `GetShapeLength` da Plataforma BusesInRio.

5.9 Identificação da Situação (Operando ou não)

Os ônibus sabidamente não operam 24 horas durante todos os dias da semana, ocorrendo paradas em estacionamentos distribuídos ao longo da cidade ou mesmo interrupções de menor duração em pontos estratégicos da cidade. Nesse período, não temos como afirmar quando os dispositivos de GPS ficam ligados enviando dados e quando não ficam, ou mesmo se no estacionamento tem ou não sinal para transmissão dos dados. Portanto, é preciso extrair esse conhecimento (saber se o ônibus está ou não operando) a partir dos dados recebidos.

Para tanto, definimos os seguintes critérios para considerar um ônibus operando:

(a) se temos menos de cinco registros recebidos e o último recebido tem até 15 minutos; ou

(b) se nenhum dos até dez últimos registros recebidos tem mais de 30 minutos em relação ao presente momento e, em relação ao último, pelo menos um deles apresenta distância maior ou igual a 200 metros (deslocamento). São testadas as dez últimas posições por essa ser a quantidade armazenada no serviço *RealTime* (veja seção 4.4) e adotado o valor de 200 metros, por esse ser superior ao deslocamento que poderia ser gerado pelo erro do dispositivo de GPS (veja seção 5.8).

Teoricamente os ônibus que atendem ao critério (b) podem ser ditos como operando boa probabilidade de acerto, sendo o critério (a) uma situação transitória

por 15 minutos. Cabe destaque que ônibus sem registro de GPS são considerados não operando, assim como, qualquer situação que não atenda um dos critérios acima.

De igual forma, podemos considerar uma linha de ônibus operando se pelo menos um dos ônibus que informam estar servindo a mesma, no registro do dispositivo de GPS, estiver operando, dado os critérios acima.

Os métodos `HowManyBusesOperating` e `HowManyRoutesOperating` fazem uso deste conceito para totalizar o número de ônibus e linhas operando. Assim como, outros métodos que oferecem algum filtro pela situação (operando x não operando).

5.10
Números do Serviço de Ônibus

Buscando uma visão geral da evolução dos serviços, implementamos uma instância do serviço *Extracion* para analisar de forma geral os dados históricos, calculando para cada dia os seguintes indicadores:

- Número de registros de dispositivo de GPS dos ônibus
- Número de ônibus distintos que enviaram registros
- Número de linhas de ônibus distintas que enviaram registros
- Número de ônibus distintos que enviaram registros e estavam operando
- Número de linhas de ônibus distintos que enviaram registros e estavam operando
- Número de registros sem informação da linha de ônibus
- Número de ônibus operando sem informação da linha de ônibus
- Mesmos sete dados acima separados por cada uma das 24 horas

Anteriormente, utilizamos a hora corrente no mesmo fuso dos dispositivos de GPS para definir se o ônibus está ou não operado no presente momento. No entanto, para uma visão histórica precisamos considerar o horário do último registro ligo como hora corrente. Os conhecimentos extraídos neste serviço são salvos no *storage*, na pasta "count", com o seguinte formato de arquivo "yyyymmdd.txt". O método `GetOverViewDay` da Plataforma BusesInRio permite acessar esses arquivos para uma data específica e o método GetOverPeriod para

um período (data início e data fim). No próximo capítulo apresentaremos o consumo deste conhecimento.

Adicionalmente, consolidamos este conhecimento em algumas visões gráficas. Da Figura 54 até a Figura 60 apresentamos a evolução histórica do total de linhas de ônibus (total x operando) para cada dia da semana. Enquanto a Figura 61 apresenta para todos os dias. Enquanto, da Figura 62 até a Figura 68, apresentamos a evolução histórica do total de ônibus (total x operando) para cada dia da semana. De igual forma, o Figura 69 inclui todos os dias da semana.

Uma conclusão interessante a partir da Figura 61 e Figura 69, é que o número de linhas apresenta uma tendência de redução mais acelerada que o número de ônibus.

A depressão observada nos gráficos entre os meses de Agosto, Setembro e Outubro de 2015 pode ser originada pelo alto número de ônibus sem informação de linha circulando no período e por uma possível reorganização das linhas da Cidade.

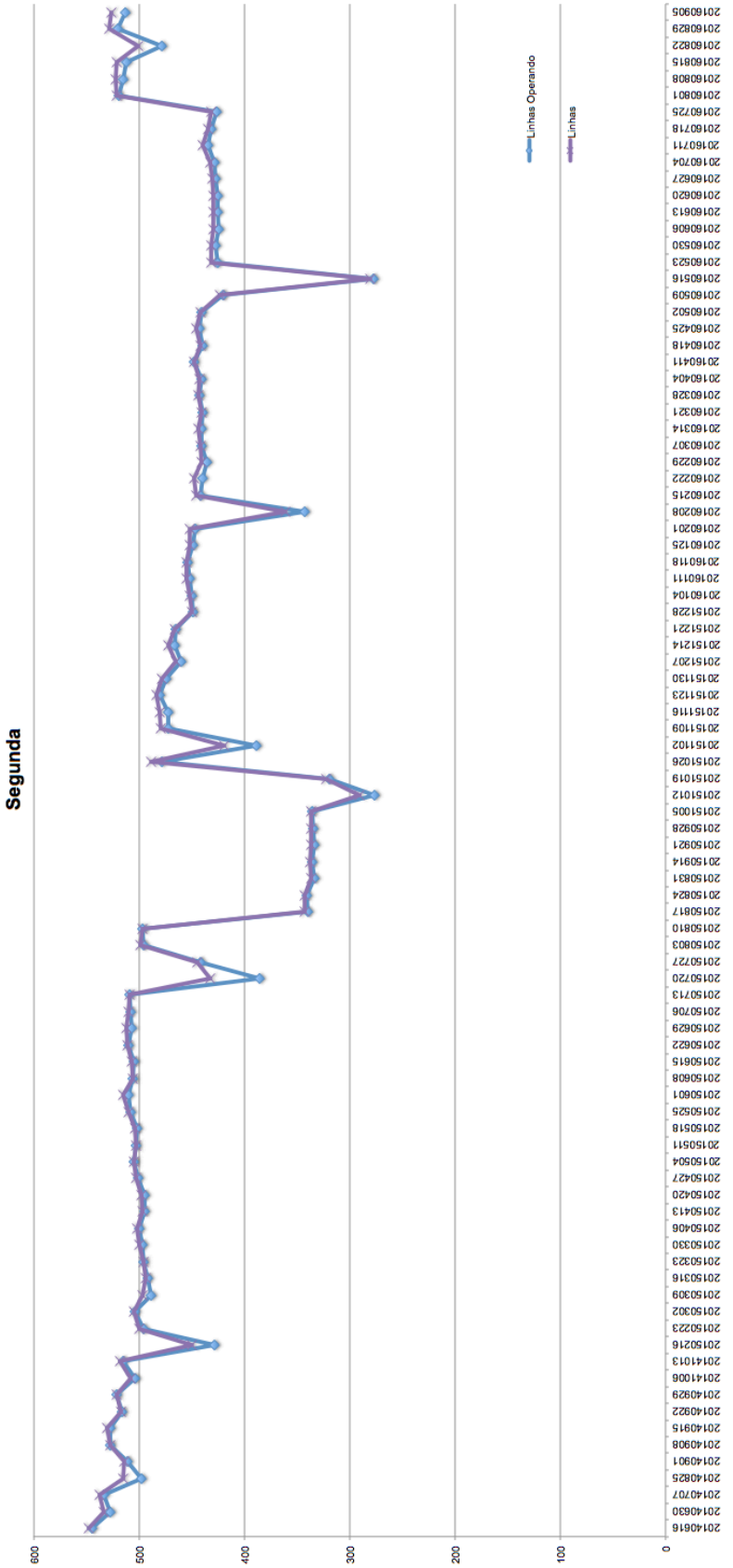


Figura 54 – Linhas Total x Operando nas Segundas-feiras

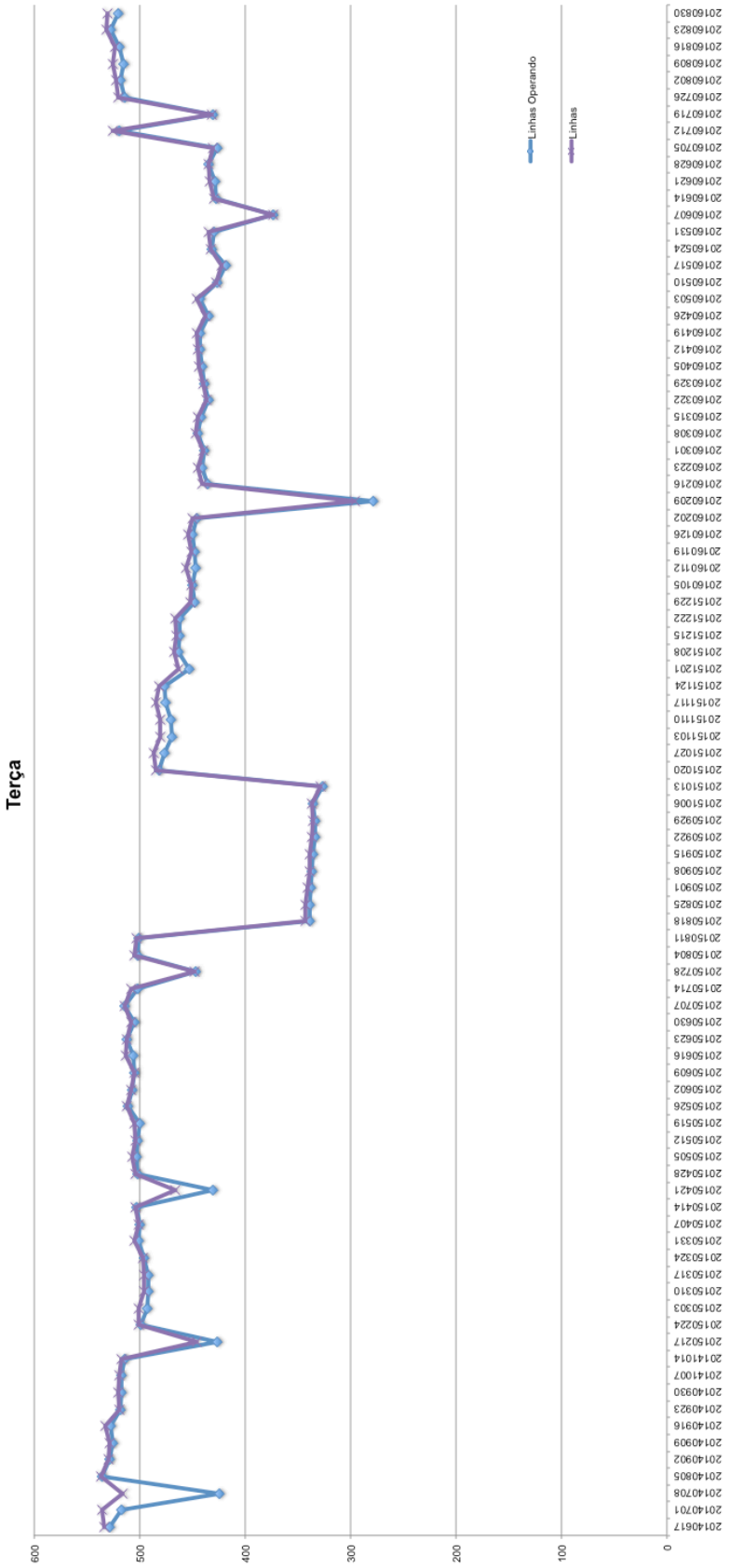


Figura 55 – Linhas Total x Operando nas Terças-feiras

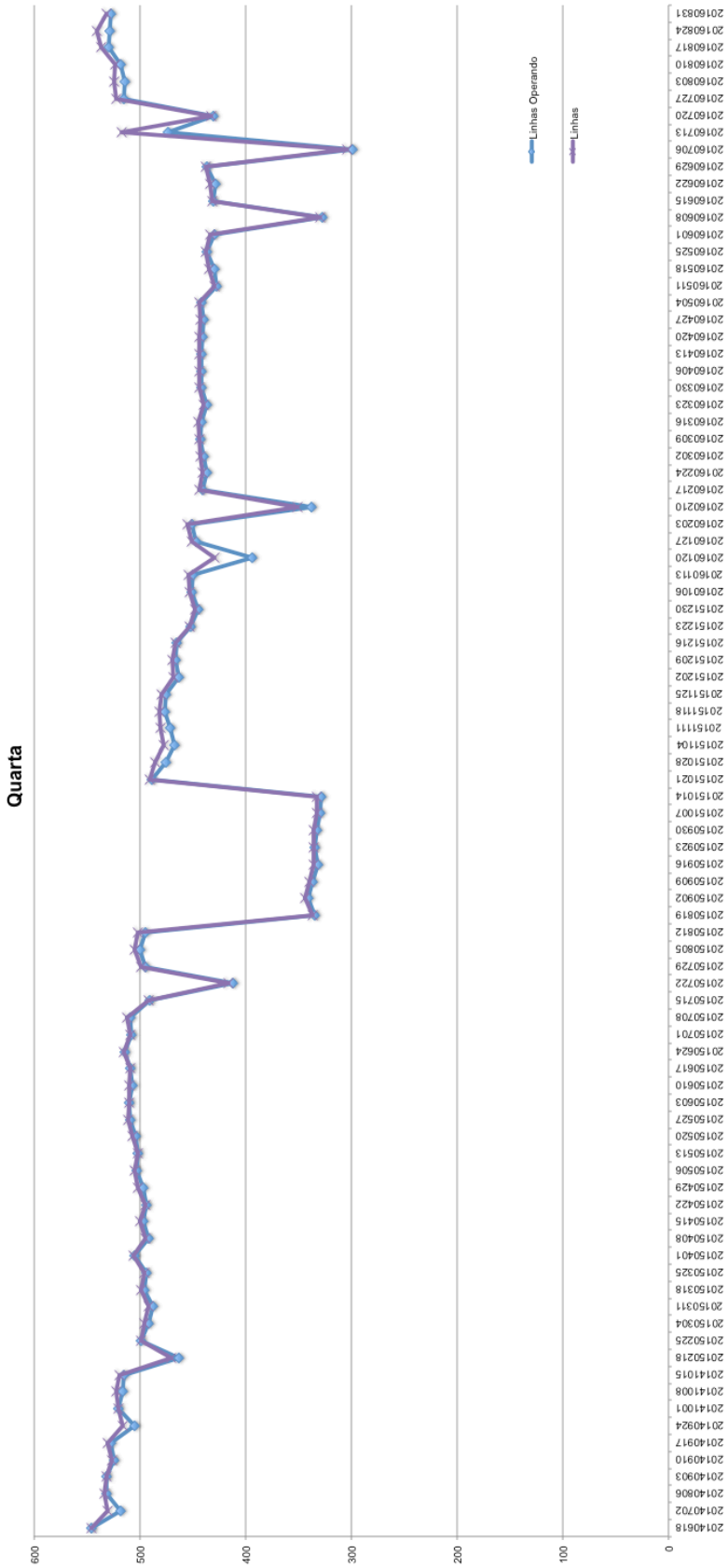


Figura 56 – Linhas Total x Operando nas Quartas-feiras

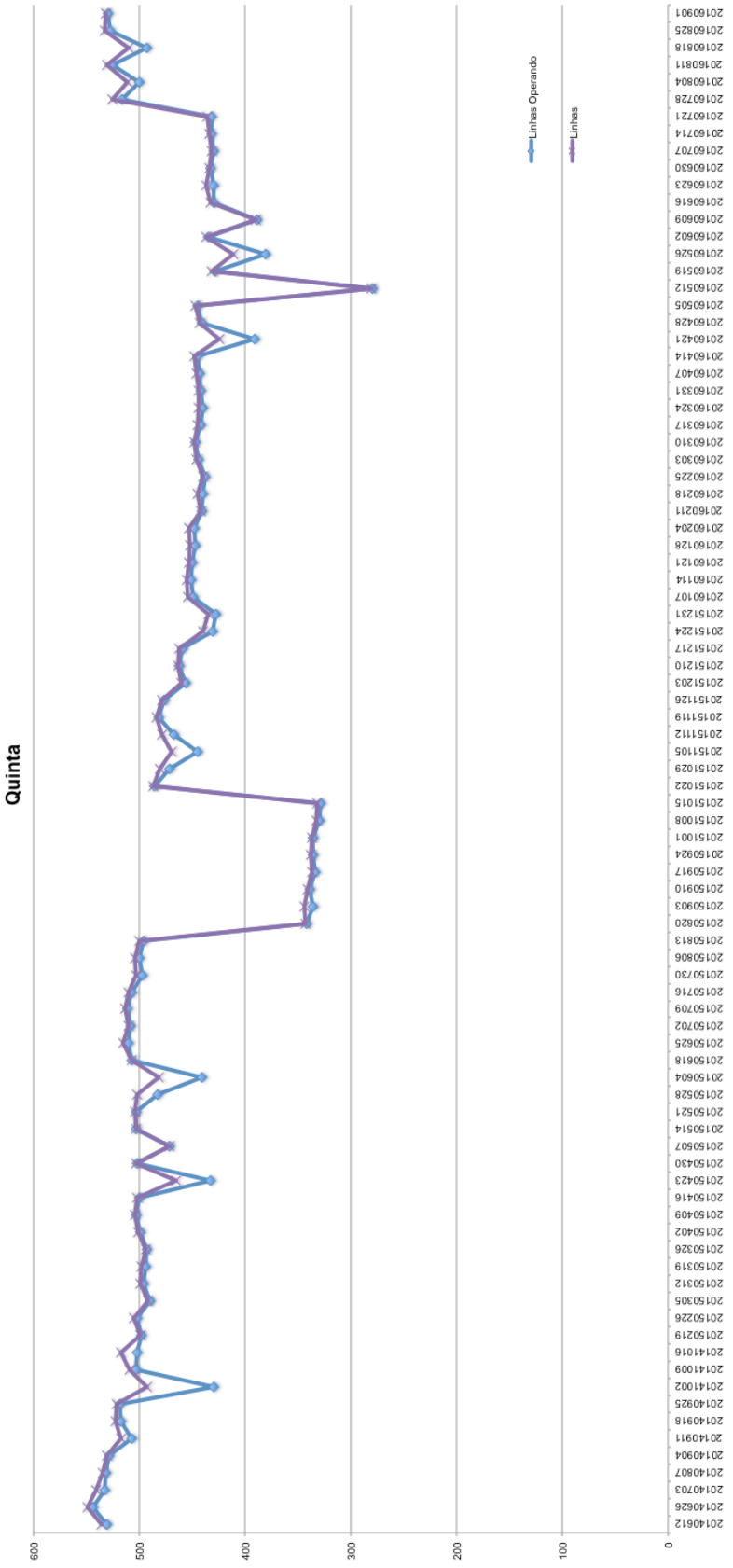


Figura 57 – Linhas Total x Operando nas Quintas-feiras

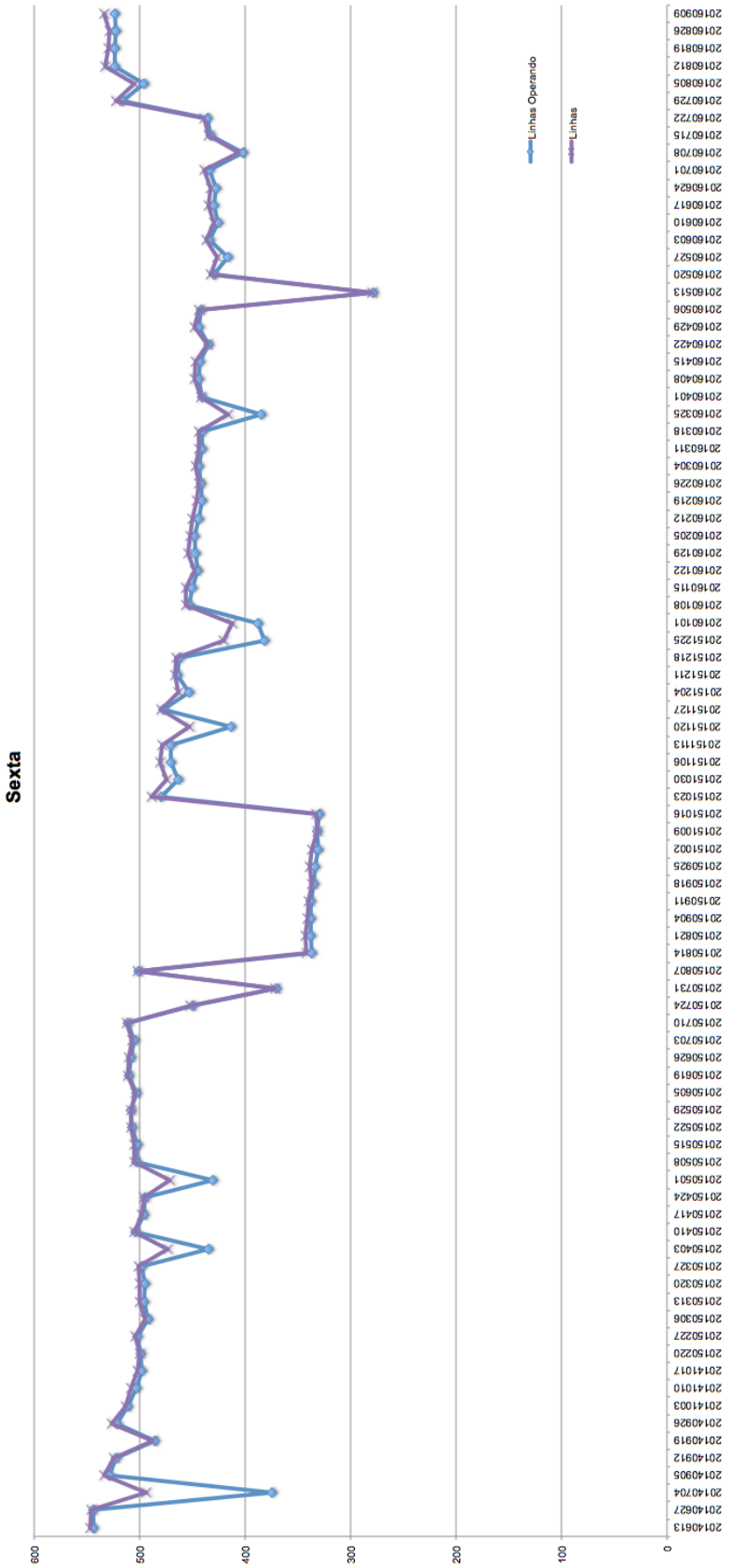


Figura 58 – Linhas Total x Operando nas Sextas-feiras

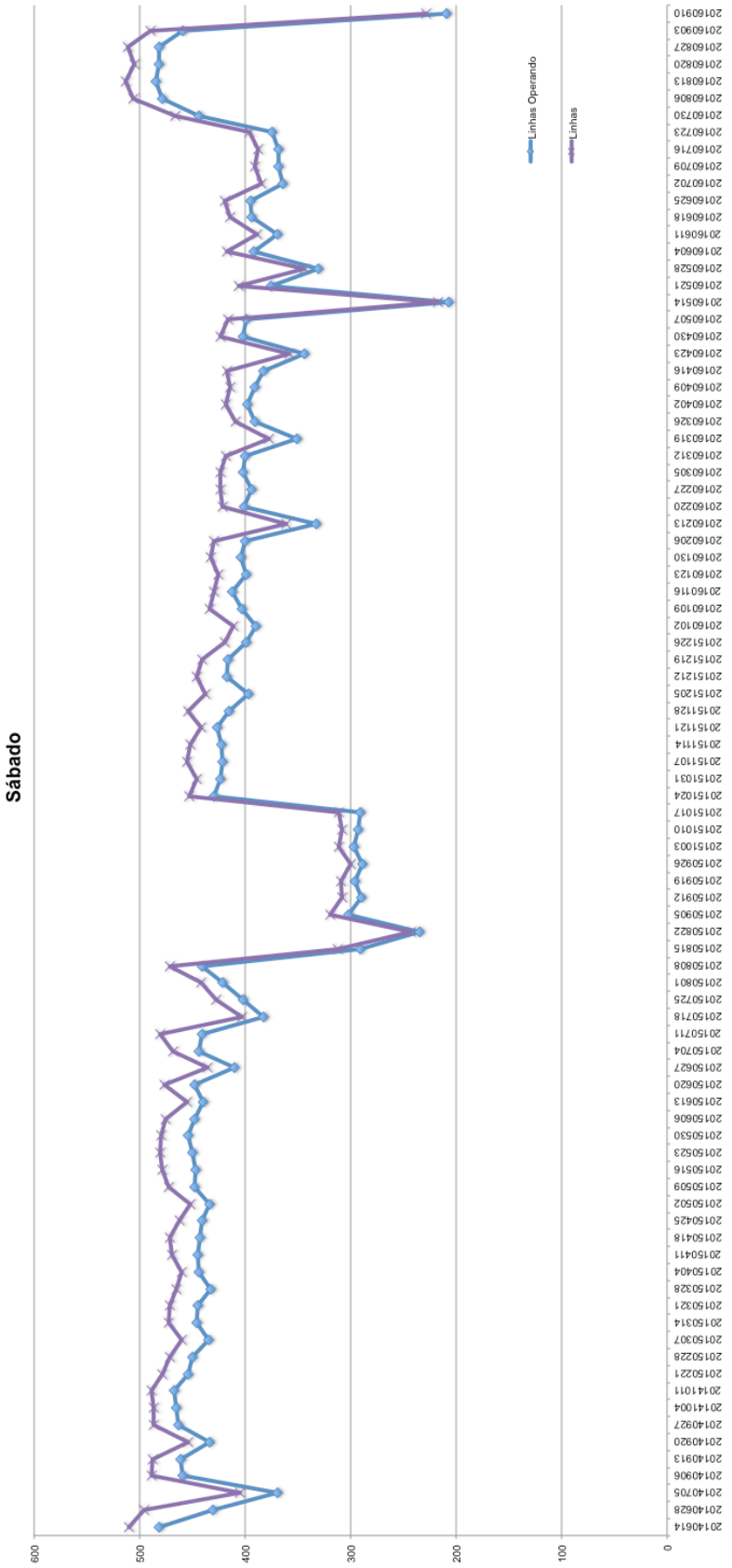


Figura 59 – Linhas Total x Operando nos Sábados

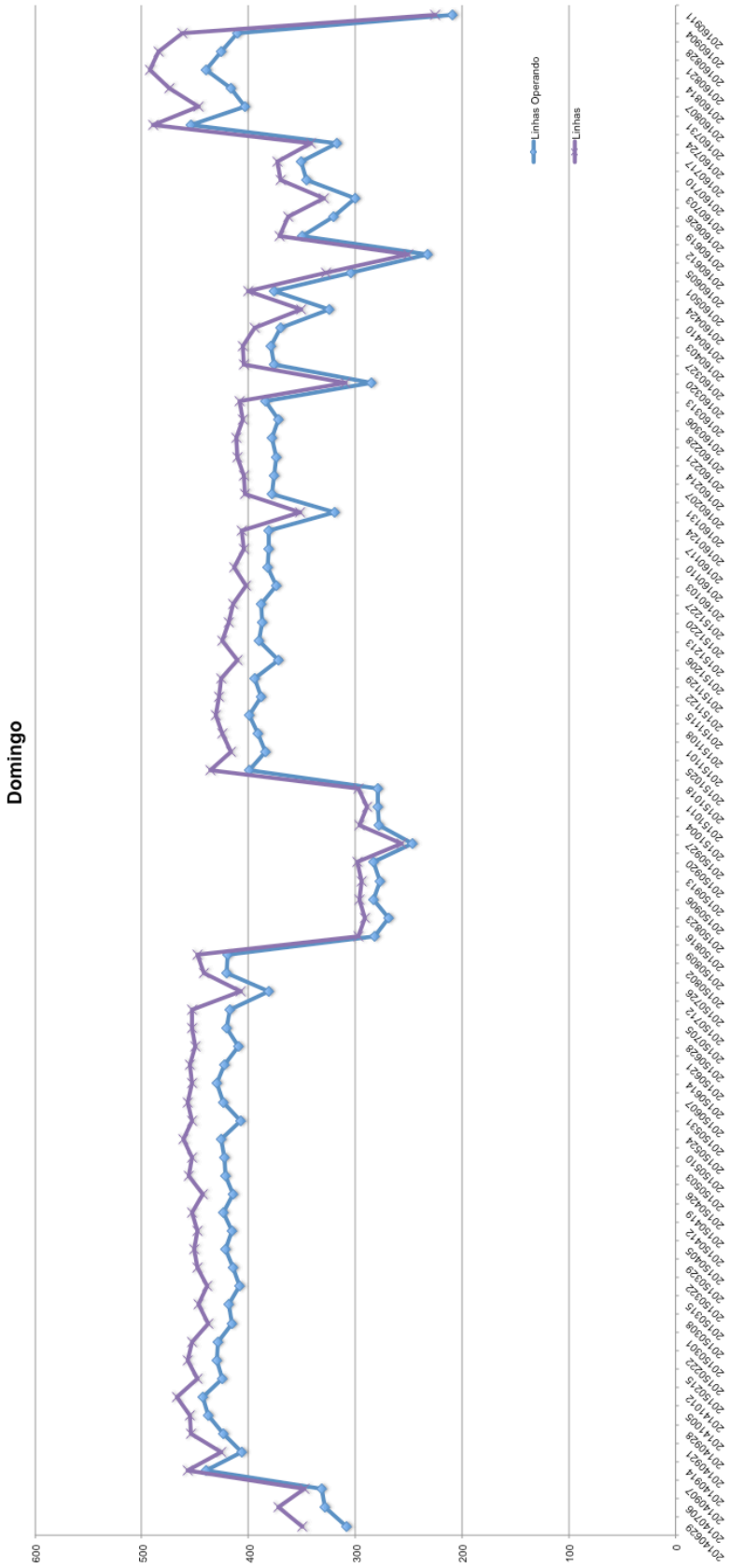


Figura 60 – Linhas Total x Operando nos Domingos

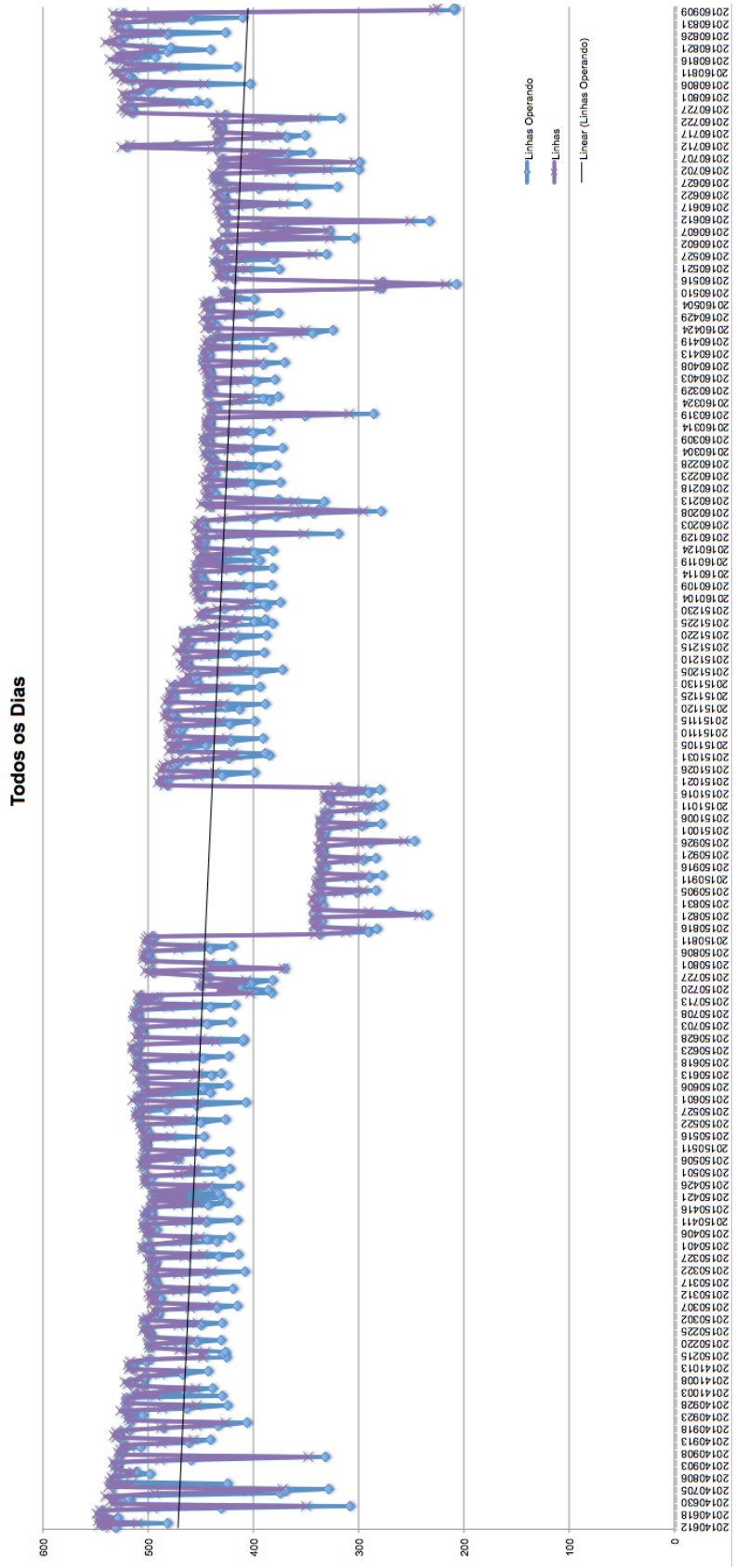


Figura 61 – Linhas Total x Operando com Linha de Tendência

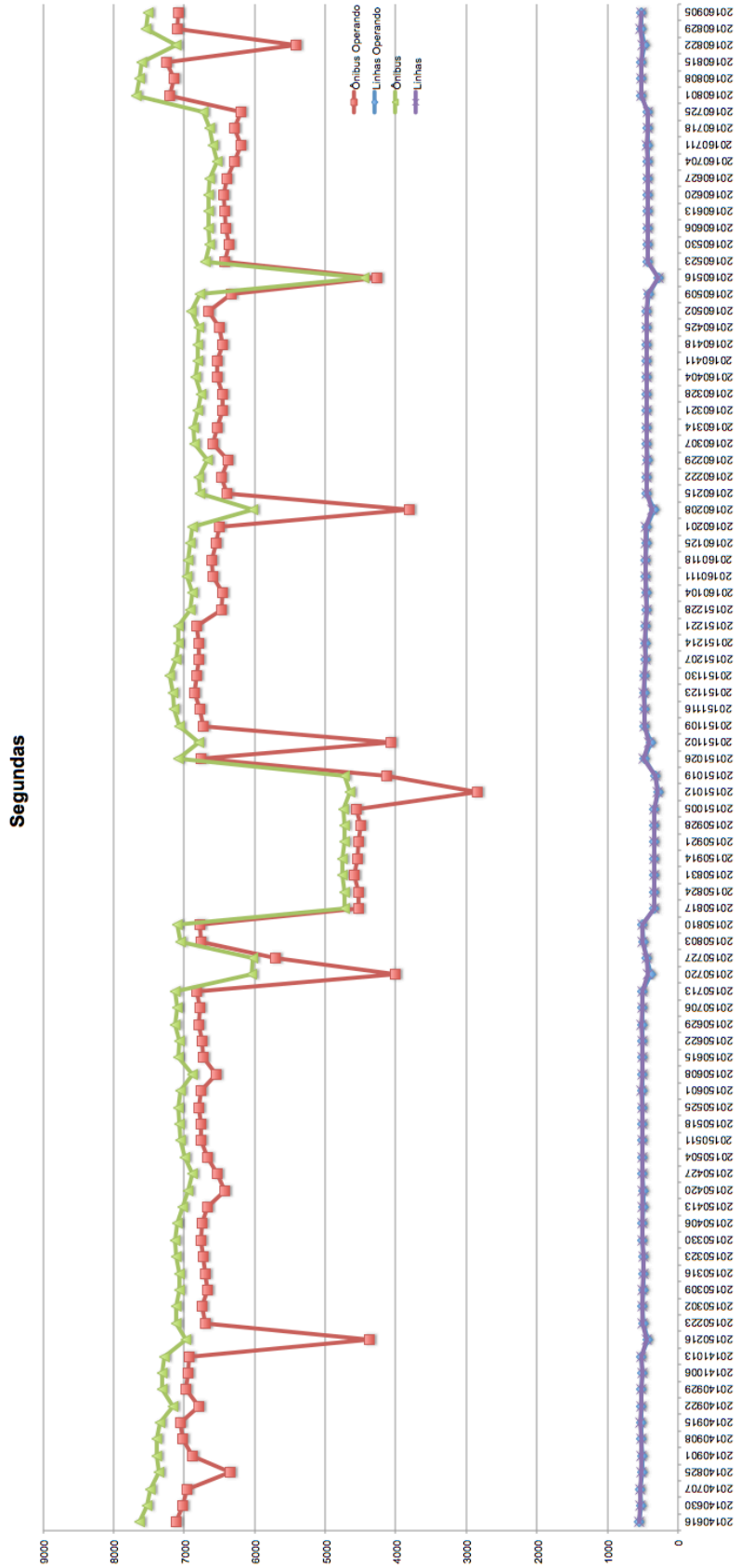


Figura 62 – Ônibus Total x Operando nas Segundas-feiras

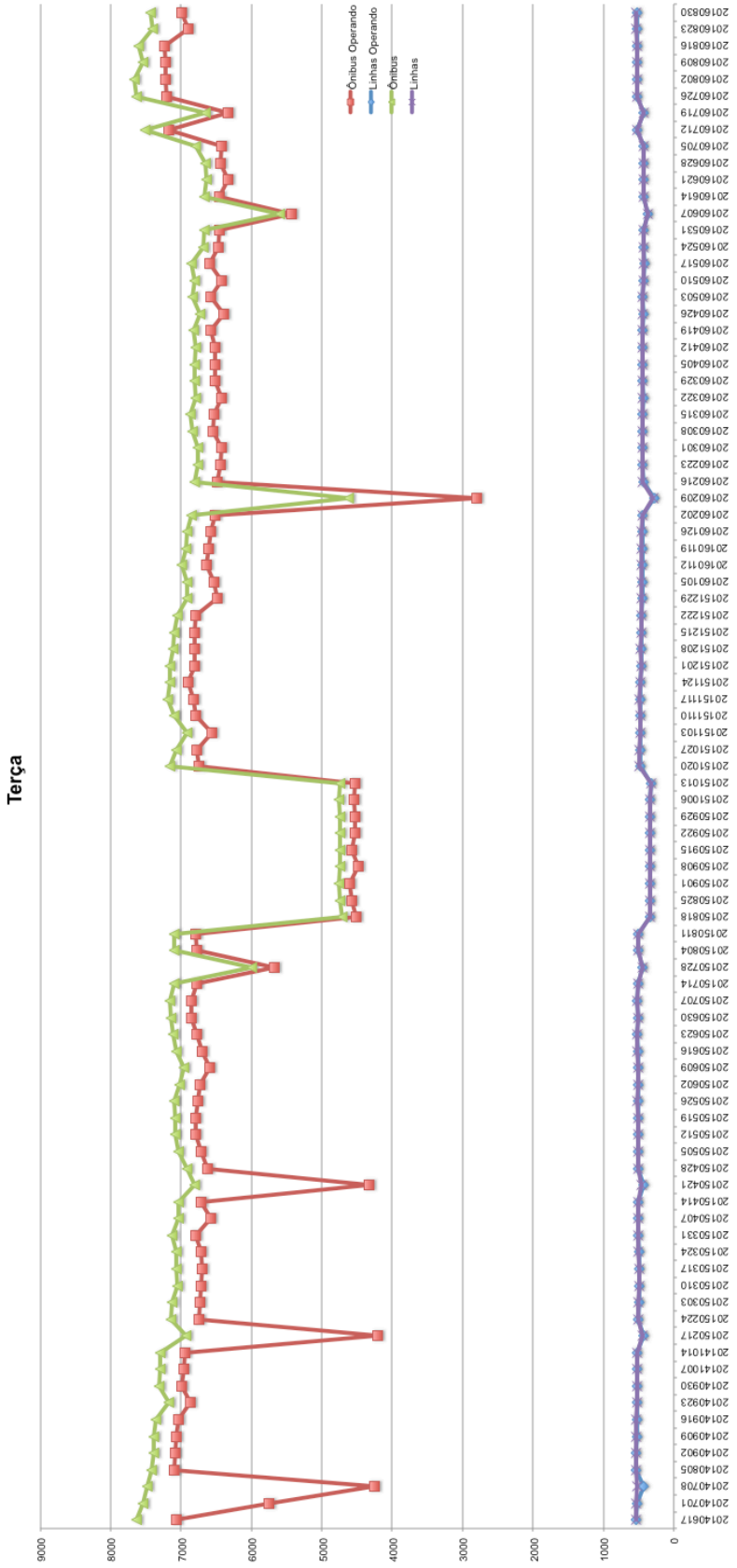


Figura 63 – Ônibus Total x Operando nas Terças-feiras

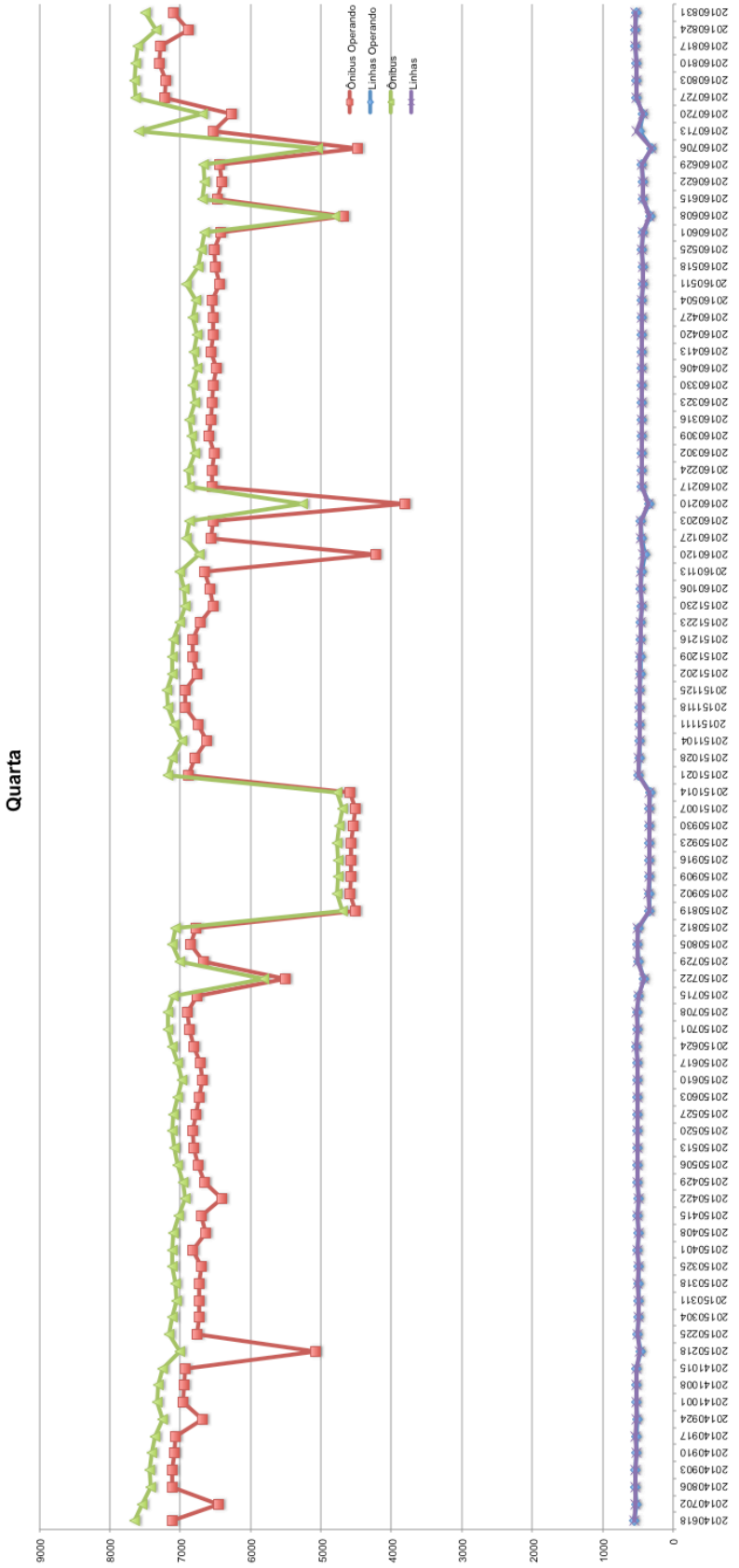


Figura 64 – Ônibus Total x Operando nas Quartas-feiras

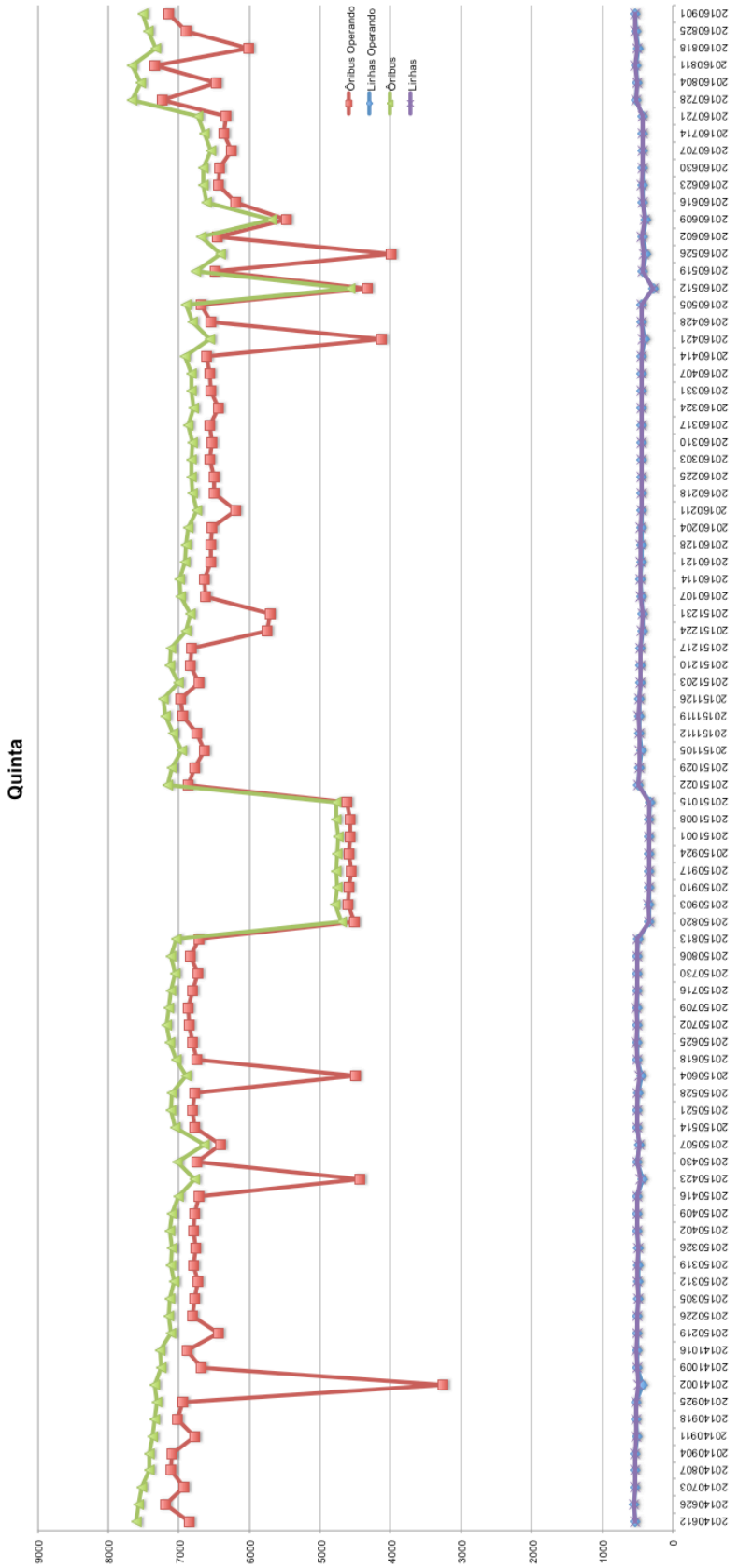


Figura 65 – Ônibus Total x Operando nas Quintas-feiras

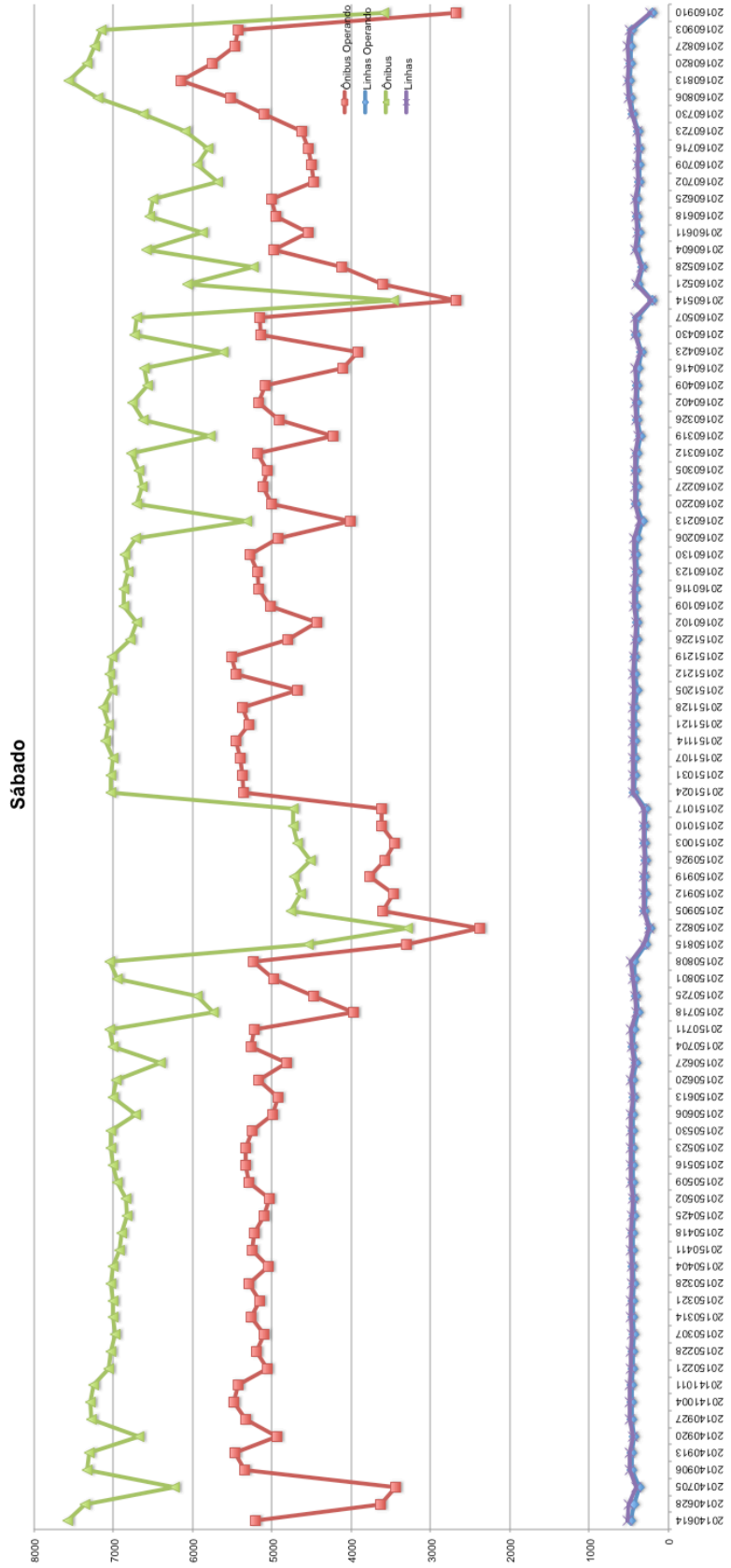


Figura 66 – Ônibus Total x Operando nas Sextas-feiras

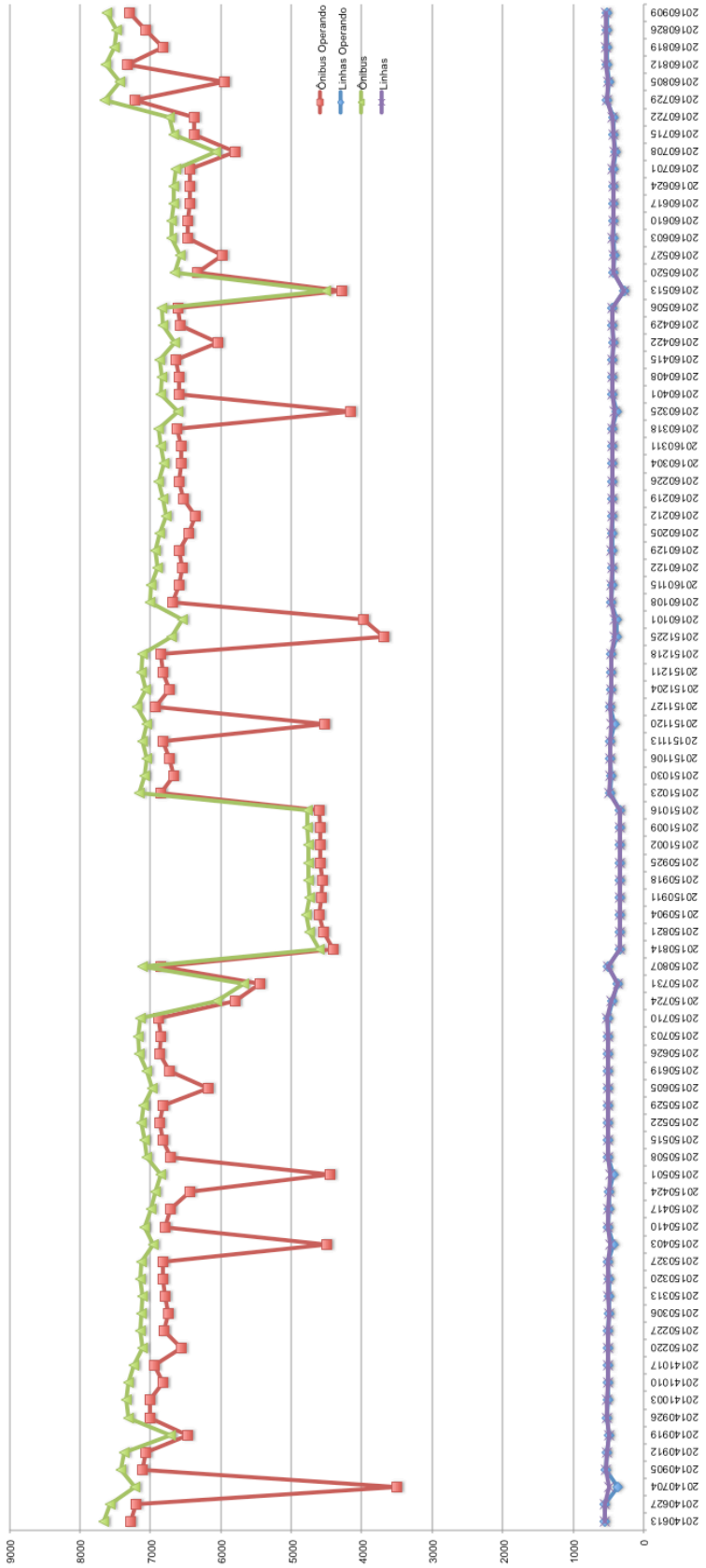


Figura 67 – Ônibus Total x Operando nos Sábados

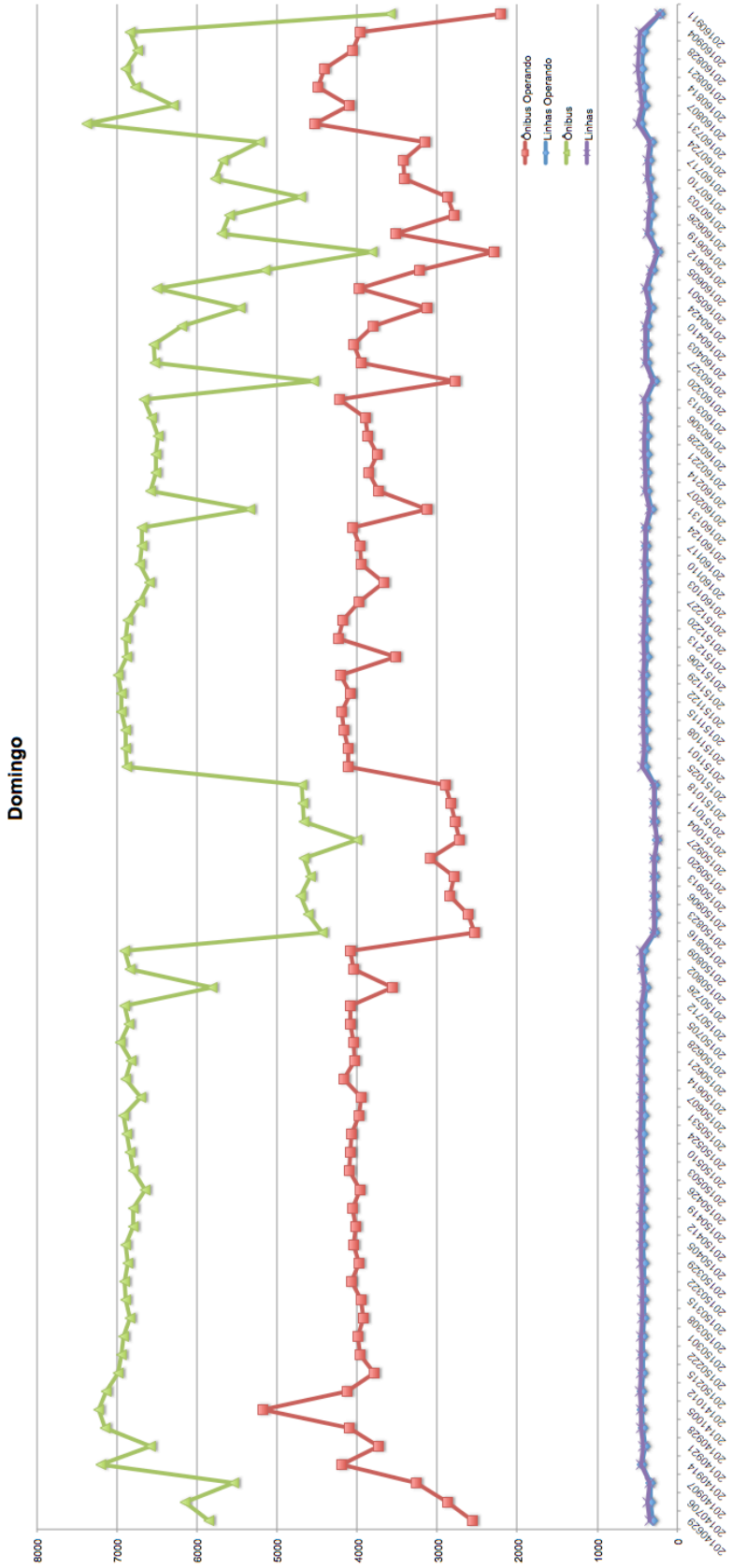


Figura 68 – Ônibus Total x Operando no Domingo

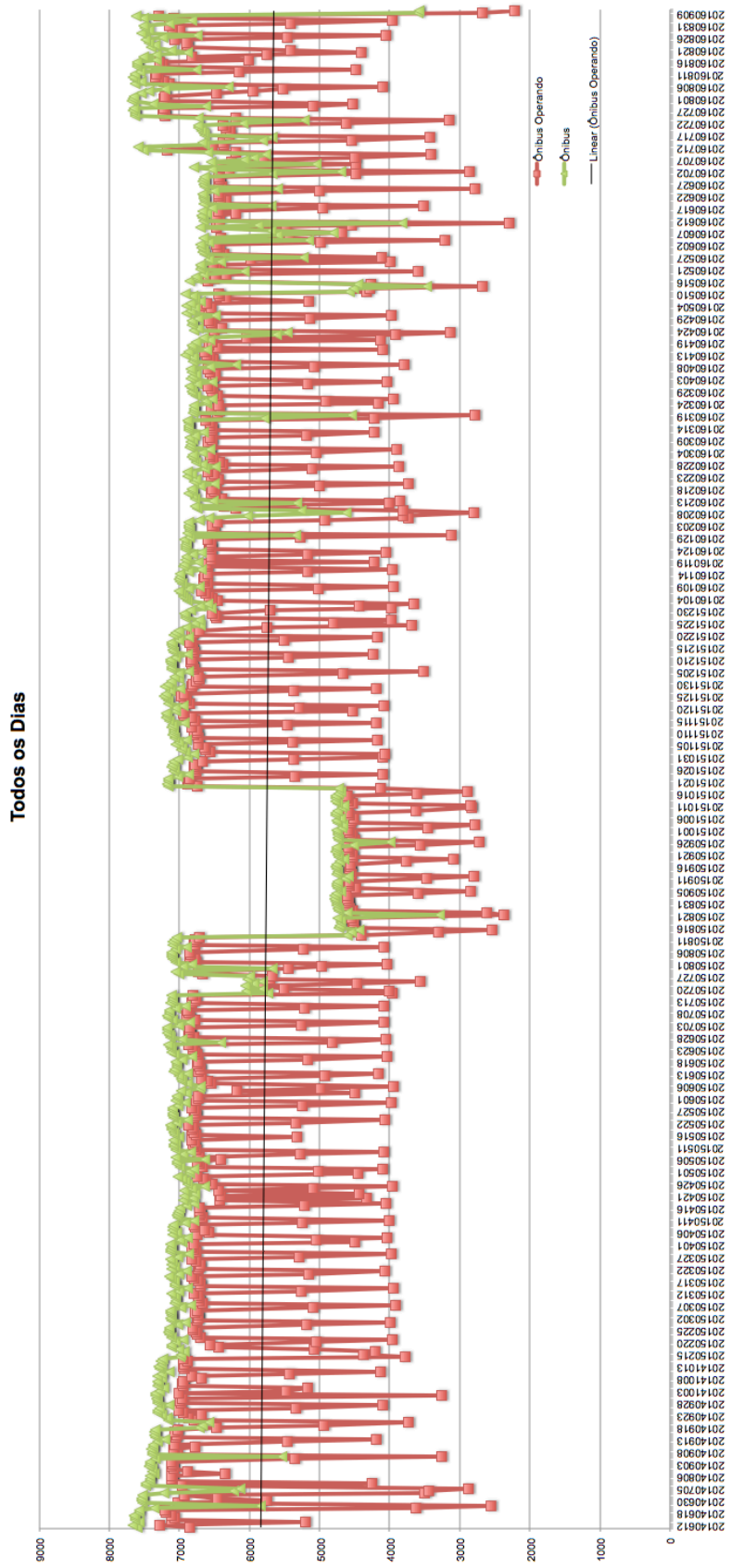


Figura 69 – Ônibus Total x Operando com Linha de Tendência

Por último, a Figura 70 demonstra a distribuição média dos ônibus operando ao longo das horas do dia.

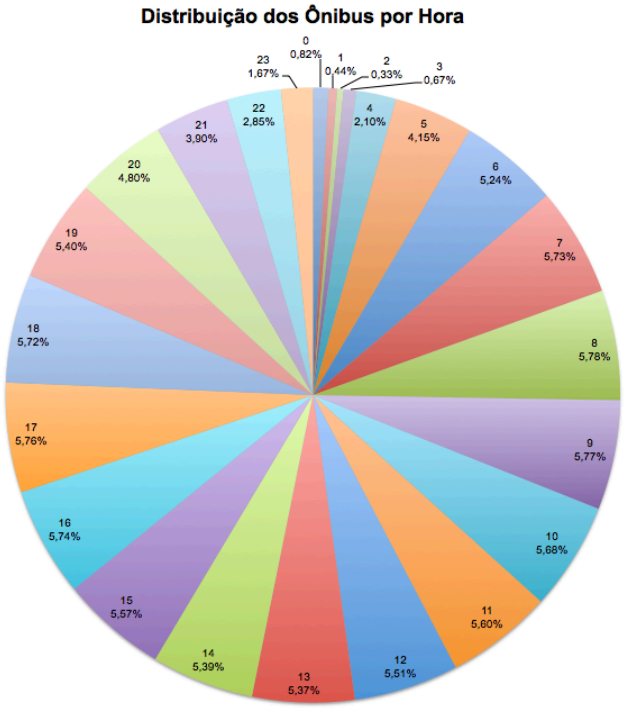


Figura 70 – Distribuição Ônibus Operando nas Horas

5.11
Extração de Rotas das Linhas de Ônibus

Como uma quantidade significativa de linhas não possui a discretização de seus trajetos, podemos utilizar dados históricos para extrair esse conhecimento. No entanto, analisando com cuidado os dados, algumas situações merecem atenção:

- O trajeto das linhas podem apresentar variações ao longo do dia e ao longo da semana, como reversões de faixas ou interdição de ruas.
- Os motoristas podem adotar percursos distintos ao trajeto da linha para cortarem caminho ou por erro.
- O percurso do motorista até o estacionamento ocorre com frequência e esse trecho essencialmente pertence ao trajeto da linha, no entanto, não ao trajeto em serviço, que é o que efetivamente interessa ao usuário.

Realizadas essas considerações, adotamos o seguinte algoritmo

(a) Selecionamos a partir dos registros do dia de ontem (e.g. 20160902.zip, arquivo compactado com todos os registros do dia 02/09/2016), os que correspondem a linha que se deseja descobrir a rota (i.e. filtro pelo identificador da linha presente no campo linha, e.g. identificador 432). Nesse processo de seleção, o décimo maior e menor valor de latitude e longitude dentre os registros selecionados são identificados, definindo-se assim uma região de interesse. Cabe comentar que são descartados alguns valores (i.e. os nove de cada extremo) para que exceções não sejam consideradas, uma vez que se supõe que existam vários ônibus próximos em coordenadas que efetivamente estão próximas a rota da linha. A Figura 71 ilustra a região e os registros selecionados para o dia e linhas citados como exemplo, ou seja, respectivamente 02/09/2016 e 432, contemplando assim, 21.046 registros (coordenadas de latitude/longitude recebidas do dispositivo de GPS dos ônibus) distintos desta linha ao longo do dia em questão.

Figura 71 – Registros selecionados da linha 432 em 02/09/2016

(b) Utilizando dados de arruamento do OpenStreetMap [80], que foram anteriormente obtidos na seção 5.1, é possível selecionar os pares de pontos que discretizam arruamentos que estão contidos na região de interesse, definida no

passo anterior. A Figura 72 apresenta esses pares, que totalizam 41.542 pares, utilizando cores randômicas, obviamente, dentro da mesma região.

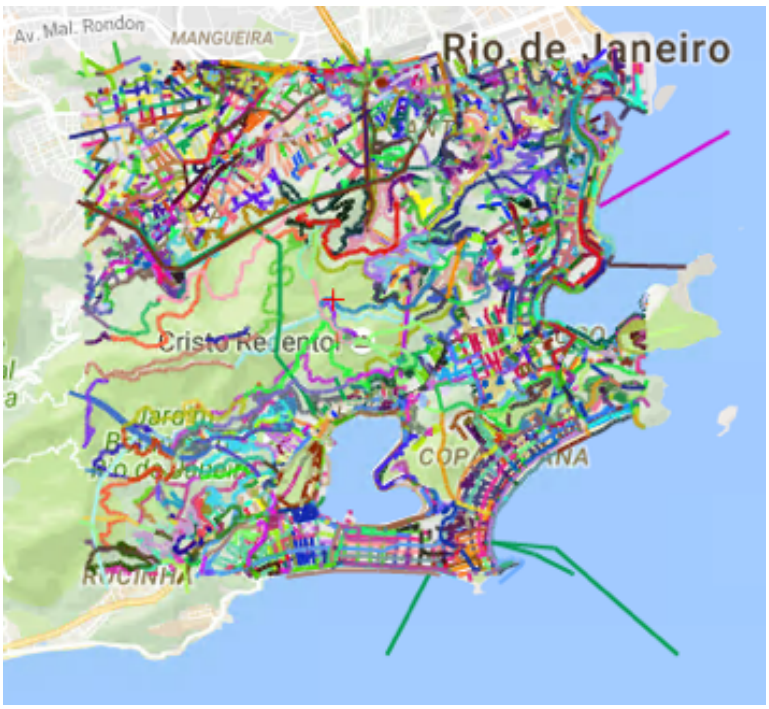


Figura 72 – Pares discretizando arruamentos dentro da região

(c) Para cada registro de GPS selecionado em (a) descobrimos o par obtido em (b) que tem menor distância ortogonal para ele e, se essa distância for inferior a 100 metros, incrementamos o contador de registros próximos deste par. Os pares que ao final possuírem no mínimo 10 registros próximos contados, são mantidos como pares candidatos para discretização da linha. Cabe registro que para evitar que para cada ponto seja necessário calcular a distância para todos os pares, realizamos uma testagem dos limites de latitude e longitude entre essas coordenadas de forma a descartar situações que não possuirão projeção ortogonal. A Figura 73 apresenta os pares que foram eleitos como candidatos para discretização da linha de identificador 432, a partir desta abordagem, obtidos utilizando-se os registros do dia 02/09/2016 selecionados anteriormente, que agora totalizam 465 pares.

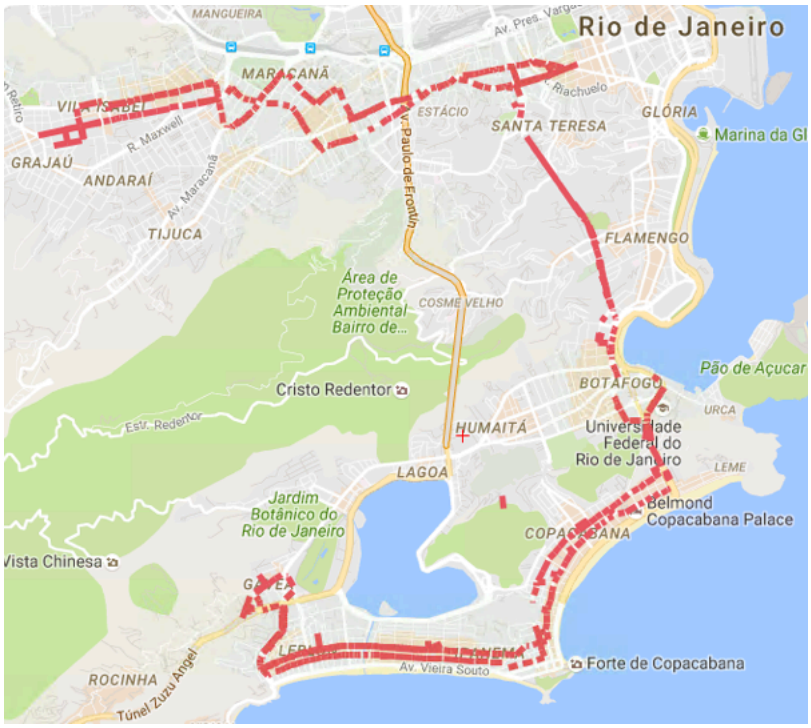


Figura 73 – Pares candidatos a discritizar a linha de ônibus

(d) Buscando descartar dentre os pares candidatos os que não são próximos aos demais, testamos a distancia dos pontos de cada par para os demais, mantendo apenas os que tem pelo menos duas coordenadas até 100 metros de distância. A Figura 74 apresenta os 458 pares que atenderam ao critério.

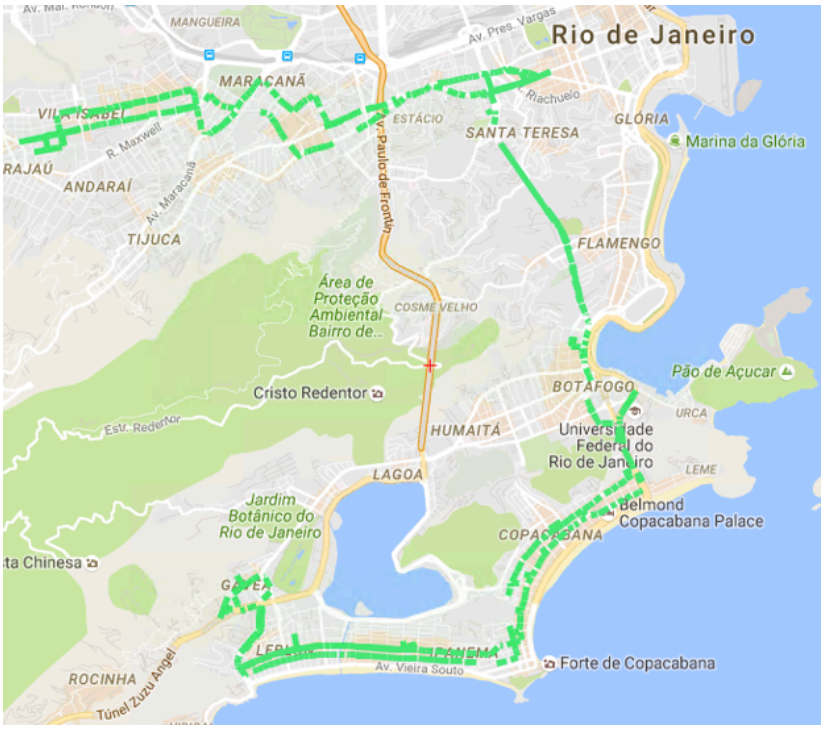


Figura 74 – Pares candidatos após novo filtro

(e) Os pares são ordenados em função da distância entre o último e primeiro ponto de cada um. Posteriormente, são conectados os que apresentam essa menor distância de até 100 metros. Os pares são quebrados e cada coordenada passa a fazer parte da descretização da linha em questão, sendo pontos repetidos descartados (possivelmente o ponto final de um par era o mesmo do inicial do próximo par). A Figura 75 apresenta a discretização final da linha 432.

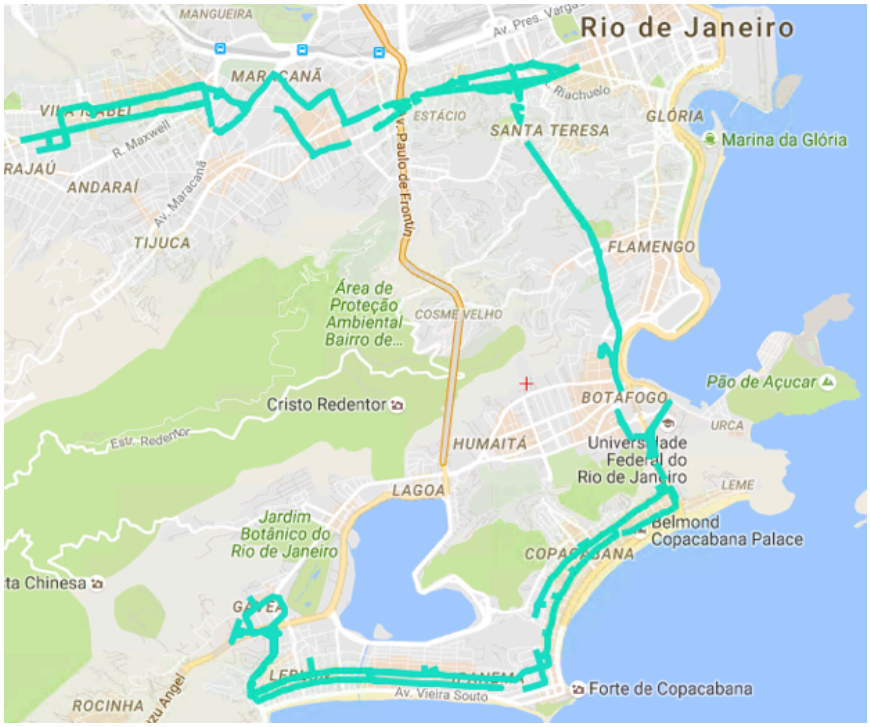


Figura 75 – Doscretização da linha 432

A aproximação em questão poderia ser aprimorada utilizando-se um maior número dados de GPS, dado que existem trechos intuitivamente pertencentes a rota, como destacado na Figura 76. Outras testagens também poderiam ser incluídas para buscar maior refinamento, como uma avaliação sequencial de pontos.

Figura 76 – Destaque de trecho que deve pertencer a rota

Não obstante a essas oportunidades de aprimoramento, podemos concluir que a discretização obtida é uma aproximação válida, a partir da observação da discretização das rotas da linha 432 informadas pela Prefeitura via arquivo CSV (veja na seção 3.1), na data de 02/09/2016, apresentada na Figura 77.

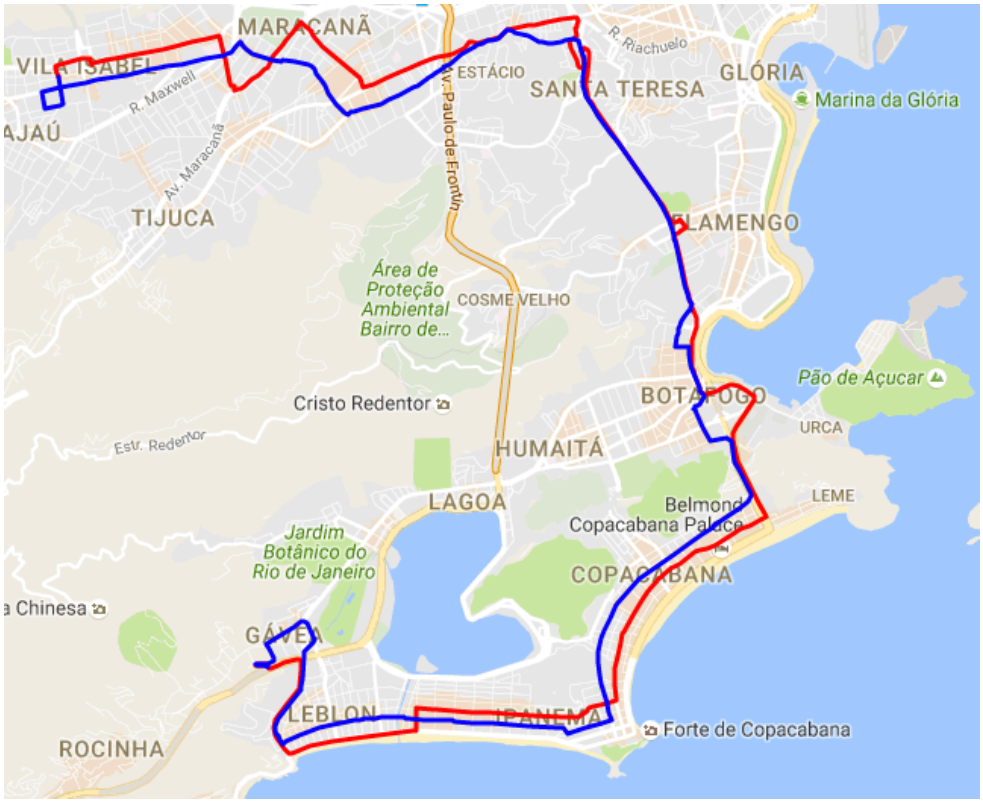


Figura 77 – Descritização da linha 432 em 02/09/2016

Comprovasse assim, mais uma vez, a utilidade da Plataforma BusesInRio para se extrair conhecimento a partir dos dados georreferenciados de mobilidade urbana, ficando sugestionada a possibilidade de melhorias neste algoritmo de extração de conhecimento para trabalhos futuros.

5.12
Exemplo de Implantação com a Plataforma

Na seção 5.3 foi apresentado o algoritmo de identificação de trajetórias de ônibus pelos segmentos monitorados. Considerando um segmento qualquer, representado pelo objeto *segment*, instanciado da classe *Segment* da Plataforma BusesInRio, que possui métodos para: verificar se um ponto georreferenciado encontra-se dentro de sua região ou mesmo próximo dos pontos que a discretizam; se uma trajetória está na mesma direção do segmento e possui um comprimento

na proporção minimamente exigida para ser válido; salvar uma trajetória sobre este segmento válida. Podemos implementar o algoritmo supracitado com a rotina abaixo.

```
VARIÁVEL GLOBAL:
// armazena em memória as trajetórias iniciadas e não concluídas de um ônibus em um
segmento, onde uma trajetória é um cojunto pontos georreferenciados em ordem
cronológica
SortedList<string, SortedList<DateTime, GeographyPoint>> Trajectories =
     new SortedList<string, SortedList<DateTime, GeographyPoint>>();


SUBROTINA (implementação a ser inserida na rotina da Figura 16)

// identificador único de um ônibus (ordem) passando pelo segmento (id)
string key = segment.Id.ToString() + "_" + ordem;

// nova trajetória é aberta se necessário
if (!Trajectories.ContainsKey(key)) {
   if (segment.InRegion(point)) {
      if (segment.InSegment(point)) {
          Trajectories.Add(key, new SortedList<DateTime, GeographyPoint>());
          Trajectories[key].Add(datetime, point);
       }
   }
// existe trajetória em aberta para o ônibus no segmento
} else {
   if (segment.InRegion(point)) {
      if (segment.InSegment(point)) {
        // adiciona ponto na trajetória em questão que está aberta
        Trajectories[key].Add(datetime, point);
      } else {
         if (segment.IsSameDirection(Trajectories[key]) {
            if (segment.Lenght(Trajectories[key]) >= (segment.LengthInMeters*0.6)){
               // salva trajetória e remove da memória
               segment.SaveTrajectory(ordem, Trajectories[key]);
               Trajectories.Remove(key);
            } else {
               // descarta trajetória da memória sem salvar, pois não tem 60%
               Trajectories.Remove(key);
            }
         } else {
            // descarta trajetória da memória sem salvar, pois outra direção
            Trajectories.Remove(key);
         }
    } else {
       if (segment.IsSameDirection(Trajectories[key]) {
          if (segment.Lenght(Trajectories[key]) >= (segment.LengthInMeters*0.6)){
             // salva trajetória e remove da memória
             segment.SaveTrajectory(ordem, Trajectories[key]);
             Trajectories.Remove(key);
          } else {
             // descarta trajetória da memória sem salvar, pois não tem 60%
             Trajectories.Remove(key);
          }
      } else {
         // descarta trajetória da memória sem salvar, pois outra direção
         Trajectories.Remove(key);
      }
   }
}
```

Figura 78 – Rotina para identificar trajetórias

6
Tomando Decisão com Dados

6.1
O Aplicativo

No capítulo 5 apresentamos como são extraídos diversos conhecimentos a partir de dados de mobilidade urbana georeferenciados utilizando a Plataforma BusesInRio apresentada. Nesse capítulo os serviços disponibilizados pela Plataforma são explorados em uma aplicação (“app”) para dispositivos móveis, que tem como objetivo subsidiar decisões diárias e informar a população e os usuários do serviço de ônibus da Cidade do Rio de Janeiro.

O app foi desenvolvido em linguagem Swift para o sistema operacional iOS, ou seja, para uso no Apple iPhone (*smartphones*). Este não armazena dados locais no dispositivo e faz o consumo dos serviços da Plataforma BusesInRio utilizando um framework SOA para iOS, o SOAEngine [83]. A Figura 79 apresenta o ícone do app, nomeado como BusesInRio, na tela inicial do dispositivo.

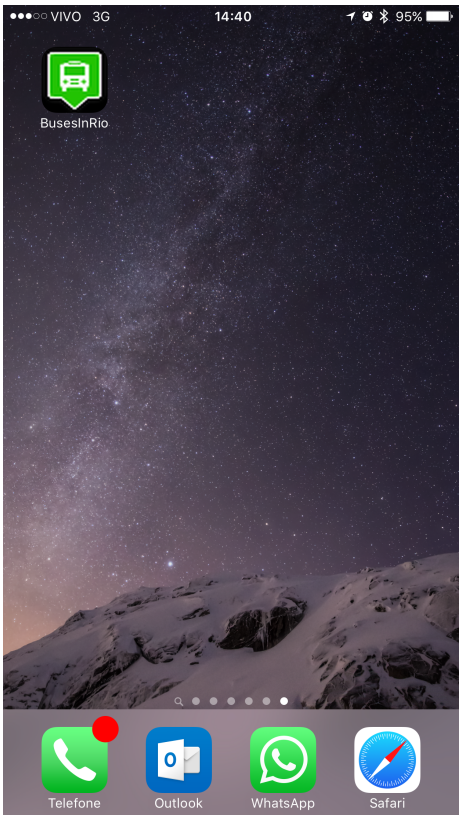


Figura 79 – Tela inicial do iPhone com ícone do BusesInRio App

6.2
Informações Básicas

As primeiros telas implementadas no app oferecem aos seus usuários informações básicas sobre o serviço de ônibus, começando abaixo com a quantidade de ônibus e linhas de ônibus operando naquele instante na Cidade.

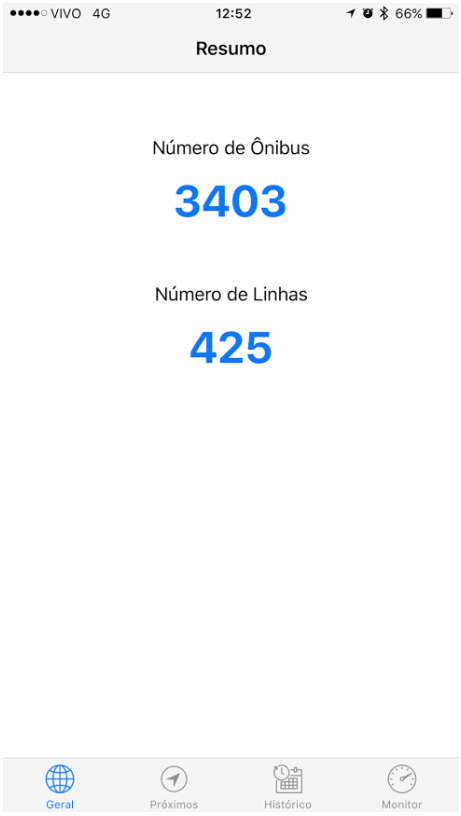


Figura 80 – Tela inicial do BusesInRio App

Como foi realizada a apresentação detalhada dos métodos da Plataforma BusesInRio anteriormente, é trivial imaginar qual método é consumido para se obter cada informação disponibilizada, portanto, não voltaremos a citar individualmente cada método consumido em cada tela.

A partir desta tela apresentada na Figura 80 é possível clicar no número de ônibus para abrir seu detalhamento ou mesmo clicar no número de linhas para abrir o detalhamento destas. Além disso, é possível mudar desta visão geral para outras visões, utilizando a barra no rodapé, no entanto, essas serão exploradas nas próximas seções.

A Figura 81 apresenta as duas telas que detalham as linhas de ônibus, sendo na esquerda a que é apresentada primeiro, contendo todas as linhas em operação naquele instante. Ao clicar no ícone verde no canto superior direito desta, é apresentada a tela da direita, contendo a lista das linhas de ônibus que não estão operando naquele momento. Clicando novamente, agora com o ícone em rosa, volta para visão das operando. Em ambas as telas é possível voltar para o contexto da Figura 80 clicando em "Resumo", assim como, aplicar filtros textuais para localizar uma linha ou mais linhas em específico.

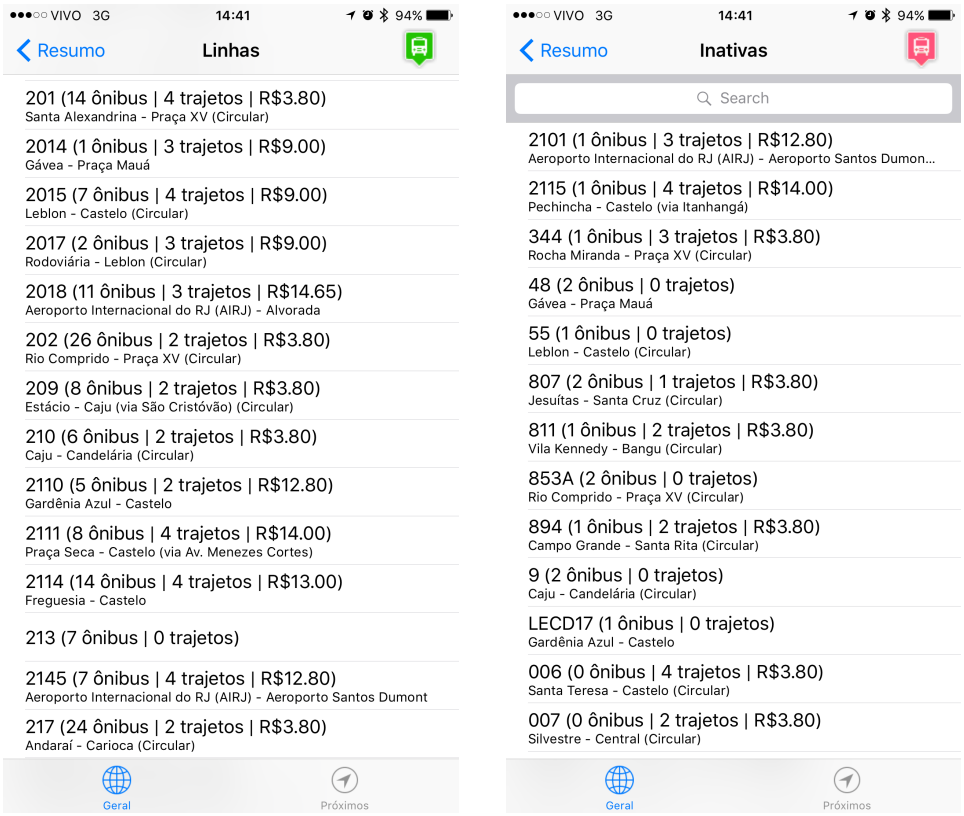


Figura 81 – Telas com lista de linhas de ônibus

Repare que nessas telas são apresentadas diversas informações sobre cada linha de ônibus, como o identificador, nome, quantidade de ônibus, quantidade de trajetos discretizados e tarifa. A lista pode ser rolada para visualizar todas as linhas na situação em questão.

Em particular, podemos escolher uma linha para obter informações adicionais, com um simples clique na mesma. Obviamente que linhas com ônibus e trajetórias discretizadas possuem mais questões a serem exploradas. A Figura 82

apresenta o detalhamento de uma linha, neste caso em particular, da linha “2018”, com seus 11 ônibus e 3 trajetos, conforme informado na figura anterior.

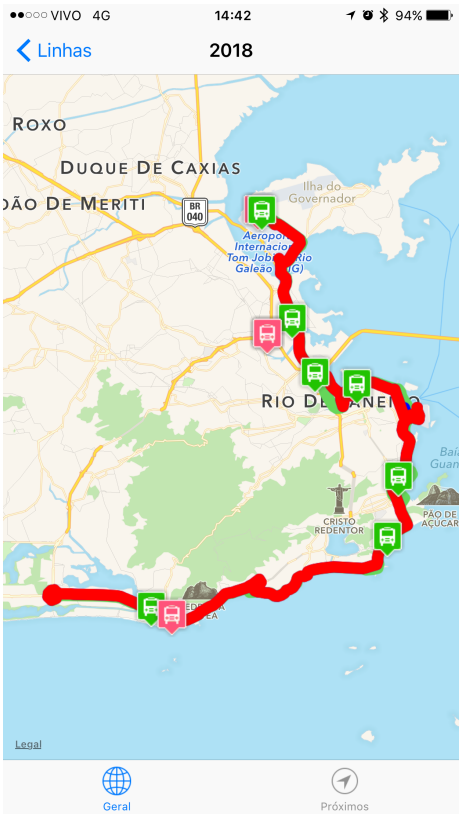


Figura 82 – Tela apresentando detalhes de uma linha

Nessa tela são apresentados os ônibus que informam servir a linha de ônibus em questão (vide cabeçalho com o identificador da linha, no caso, a “2018”), sendo em verde os que estão operando e em rosa os não estão operando. A posição exibida é a do último registro recebido do disposto GPS de cada ônibus. Os pontos que discretizam os trajetos (rotas) da linha são apresentados conectados, formando uma linha colorida para cada trajeto. O botão no canto superior esquerdo – Linhas – permite voltar para lista da Figura 81. O número da linha de ônibus fica em destaque no cabeçalho para melhor localização do usuário.

Nessa visão não é possível ver todos os ônibus e trajetos, mas aproximando se tornará perceptível a existência de ônibus e trajetos sobrepostos. Essa sobreposição é muito comum dada a existência de trajetos de ida e volta pelas mesmas ruas ou ruas muito próximas e pela proximidade de ônibus nas garagens. De qualquer forma, esses detalhes ficarão evidente ao explorarmos a linha nas figuras que se seguem.

A Figura 83 apresenta o *pop-up* exibido ao clicarmos em qualquer um dos ônibus do mapa. Nele consta a ordem do ônibus (i.e. número que consta na lateral do veículo para identifica-lo de forma única), a data e hora na qual o registro da posição foi enviado pelo dispositivo de GPS do ônibus, além de um botão para se obter informações adicionais.

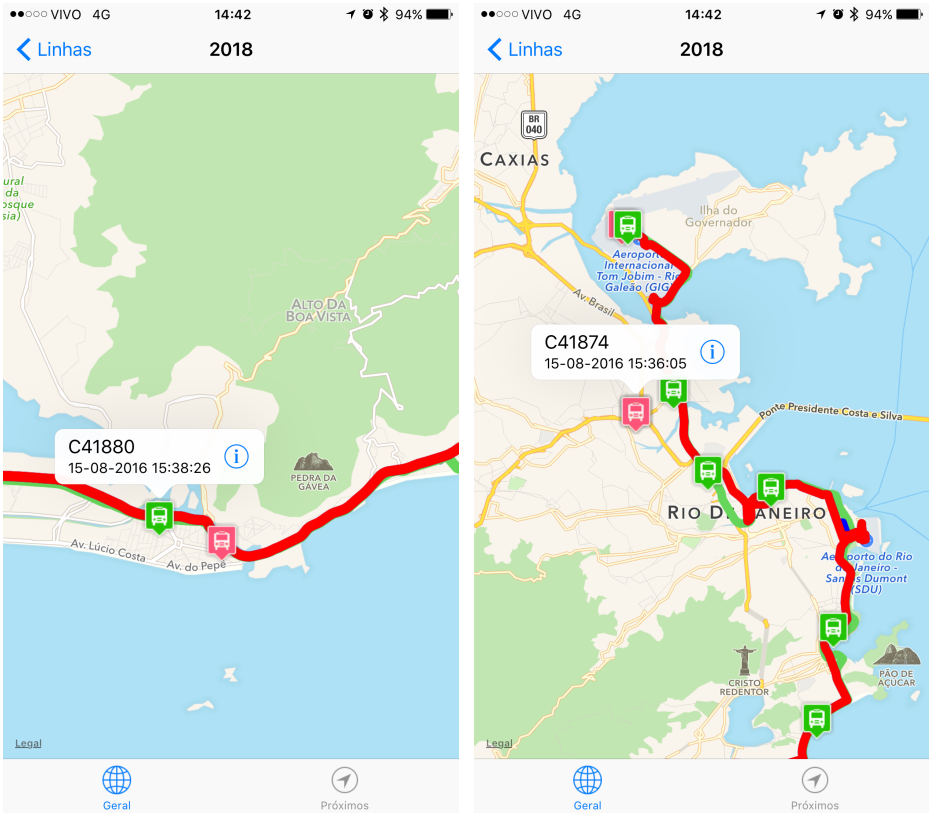


Figura 83 – Telas apresentando um ônibus da linha selecionado

Repare que nessas visualizações, em especial na tela da direita, é possível observar partes pequenas de outros trajetos em cores azuis e verde, além do trajeto em vermelho no primeiro plano. A diferença visual para as telas da Figura 82 devem-se aos recursos de zoom, mover e girar do mapa, lembrando que ambos são da mesma linha, a "2018", conforme consta no cabeçalho.

Ao clicar no ícone de informação (veja Figura 83), os ônibus em rosa e verde são retirados do mapa e são apresentados os 10 últimos registros enviados pelo dispositivo de GPS do ônibus selecionado ou os que existirem. O registro mais recente recebe o número 1, o anterior 2 e assim sucessivamente até 4, depois os registros passam a ser todos em cinza sem numeração. A Figura 84 apresenta as telas de informações dos ônibus selecionados na figura anterior, os de ordem "C41880" (esquerda) e "C41874" (direita). Outra vez foram utilizados os recursos

de manipulação do mapa para melhor a visualização, sendo mantida os trajetos da linha de ônibus em colorido.

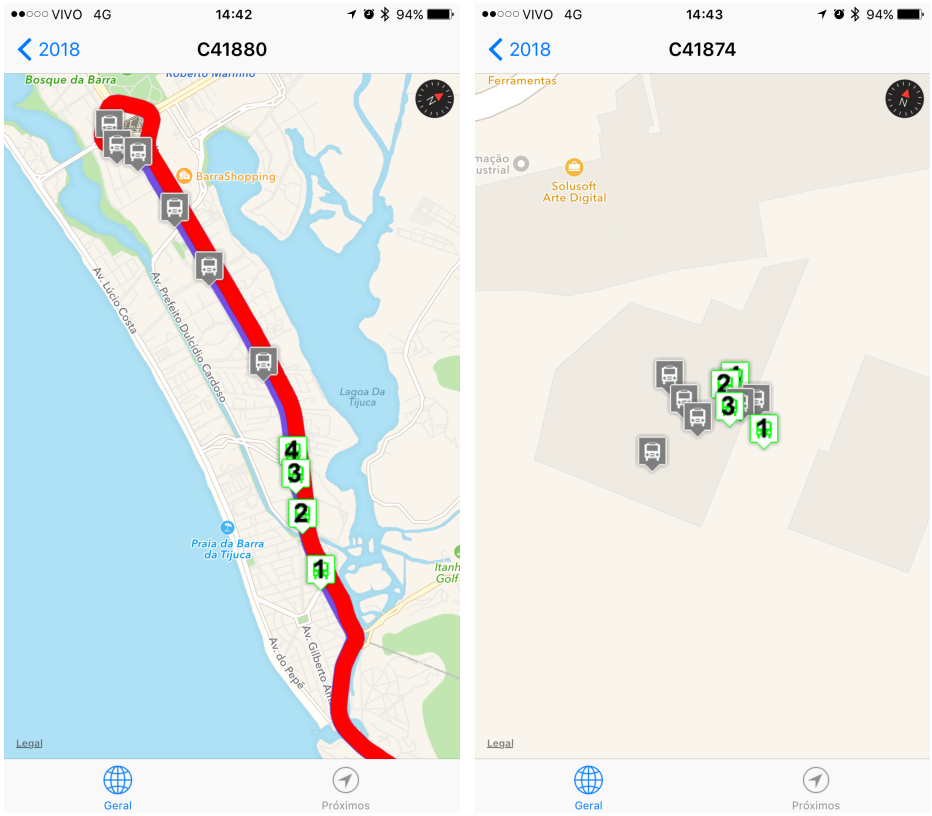


Figura 84 – Telas com informações dos ônibus selecionados

O botão no canto superior esquerdo – 2018 – permite voltar para visualização dos detalhes desta linha, conforme Figura 83. A ordem do ônibus fica em destaque no cabeçalho para melhor localização do usuário.

O ônibus “C41880“ está claramente se deslocando ao longo da linha, enquanto o ônibus ”C41874” apresenta variação insignificante entre um registro e outro - aplicamos um zoom considerável para reparar uma variação de posição entre os registros, natural da imprecisão do GPS, o que confirma a informação inicial de que ele não está operando.

Clicando em um destes registros é possível ainda obter informações adicionais sobre o mesmo, mas comentaremos essas informações adiante, uma vez que são as mesmas na análise de um ônibus independente da linha que esteja servindo, que passaremos a tratar.

A Figura 85 apresenta as duas telas que detalham os ônibus, sendo na esquerda a que é apresentada primeiro, contendo todos os ônibus em operação

naquele instante. Lembrando que essa tela pode ser aberta através do clique no número de ônibus operando da tela inicial do app (veja Figura 79).

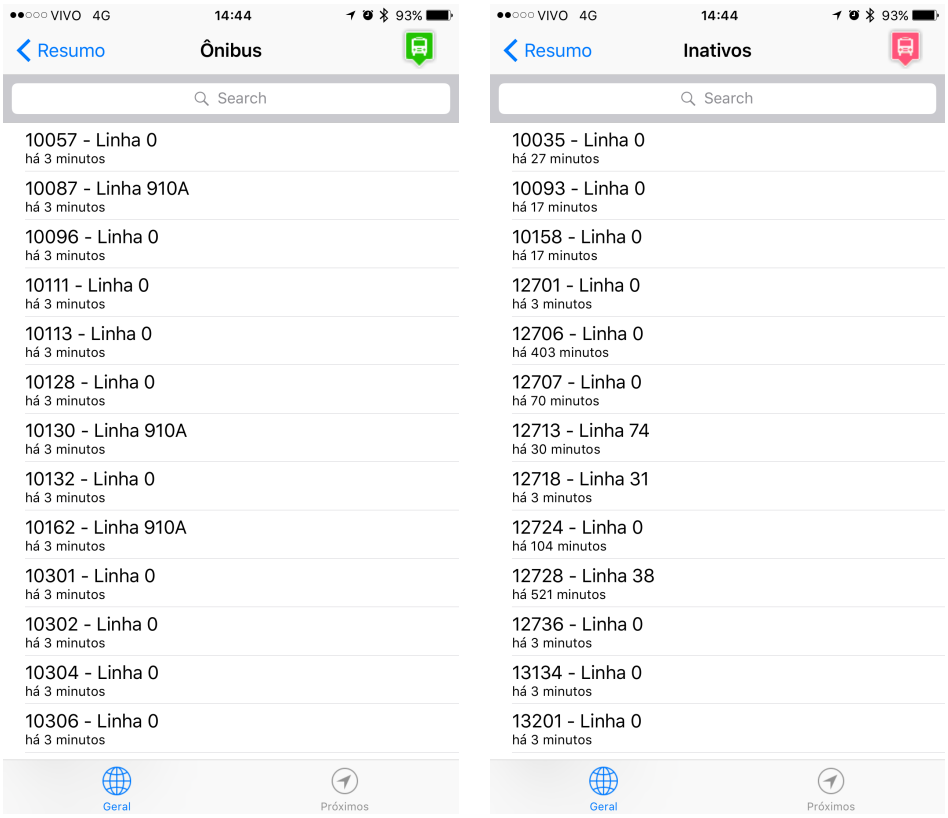


Figura 85 – Telas com lista de ônibus

O funcionamento das telas da Figura 85 e Figura 81 são análogos. É possível filtrar ou selecionar um ônibus para detalhamento. Além de filtrar e trocar entre a visualizações de ônibus operando e não operando. Nessa tela é apresentado como informação adicional o número da linha para o qual o ônibus está em serviço e zero se a informação é desconhecida. Também é apresentado o atraso do último registro recebido do dispositivo de GPS deste ônibus em relação ao momento da abertura da tela.

A Figura 86 apresenta exemplos de telas com a seleção de ônibus da Figura 85, no caso os ônibus de ordem “10130” e “12707”, estando o primeiro operando e o segundo não. Essas telas tem o mesmo comportamento e recursos da visualização de informações de um ônibus a partir da linha, visualizada na Figura 84. Mais uma vez a ordem dos ônibus é apresentada no cabeçalho.

Repare que o primeiro ônibus (“10130”) informa a linha para a qual está servindo, Linha 910A, na Figura 85, mas esta não possui pontos discretizando

seus trajetos, portanto as linhas não podem ser desenhadas, como vemos na Figura 86. Da mesma forma, o segundo ônibus (“12707”) não informa nenhuma linha, como também é possível visualizar na Figura 85.

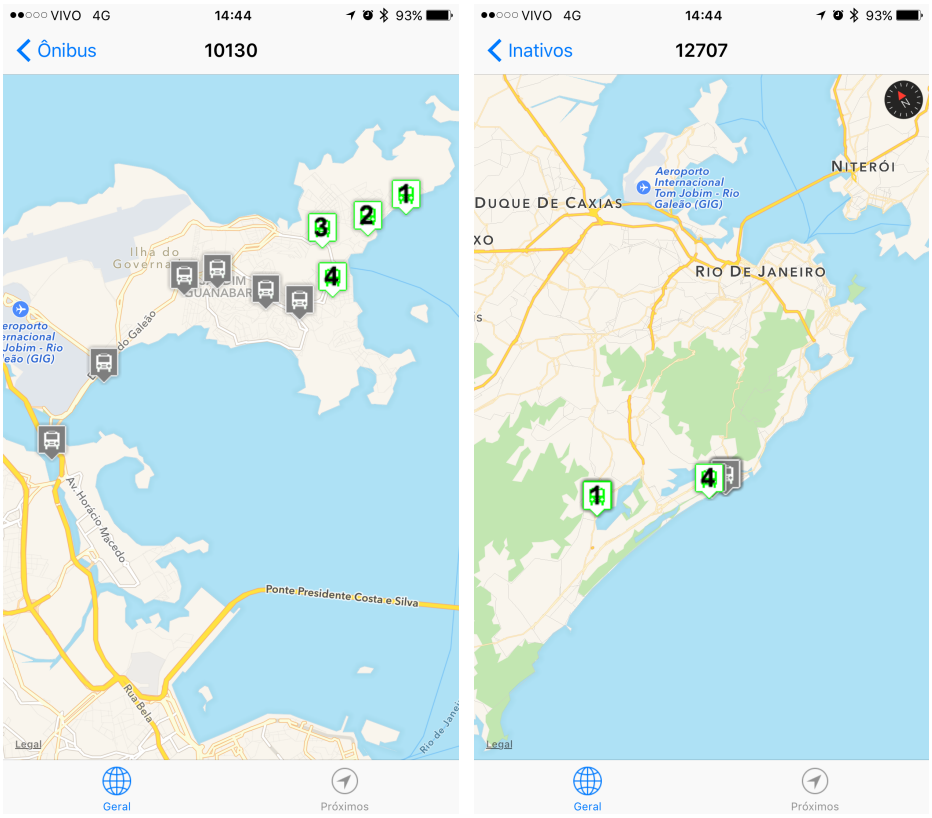


Figura 86 – Telas com visualização de ônibus operando e não

O percurso visualizado na esquerda está coerente com a situação de operando do ônibus. Enquanto o ônibus da esquerda não está operando pelo fato do último registro do dispositivo de GPS deste ônibus ter mais de 70 minutos de atraso, portanto, é possível que o dispositivo tenha sido desligado.

Essa tela tem comportamento análogo à visualização de informações de um ônibus a partir da visualização de detalhes da linha de ônibus, anteriormente ilustrada. Em ambos os casos é possível clicar em um registro deste ônibus para obter maiores detalhes.

A Figura 87 ilustra na esquerda o pop-up apresentado ao clicar em um registro de um ônibus. São exibidas a hora do registro e a distância linear (metros) e temporal (segundos) deste ponto para o anterior. Lembrando que a distância linear não é necessariamente a distância percorrida pelo ônibus. Além disso, essa pequena distância não seria suficiente para tornar o ônibus ativo, podendo tratar-se apenas da imprecisão esperada do dispositivo de GPS do veículo.

Na tela da direita é apresentado a mensagem exibida ao se clicar no ícone de informação deste *pop-up* da esquerda. O mesmo contêm a coordenada (latitude e longitude) do registro e sua data e hora.

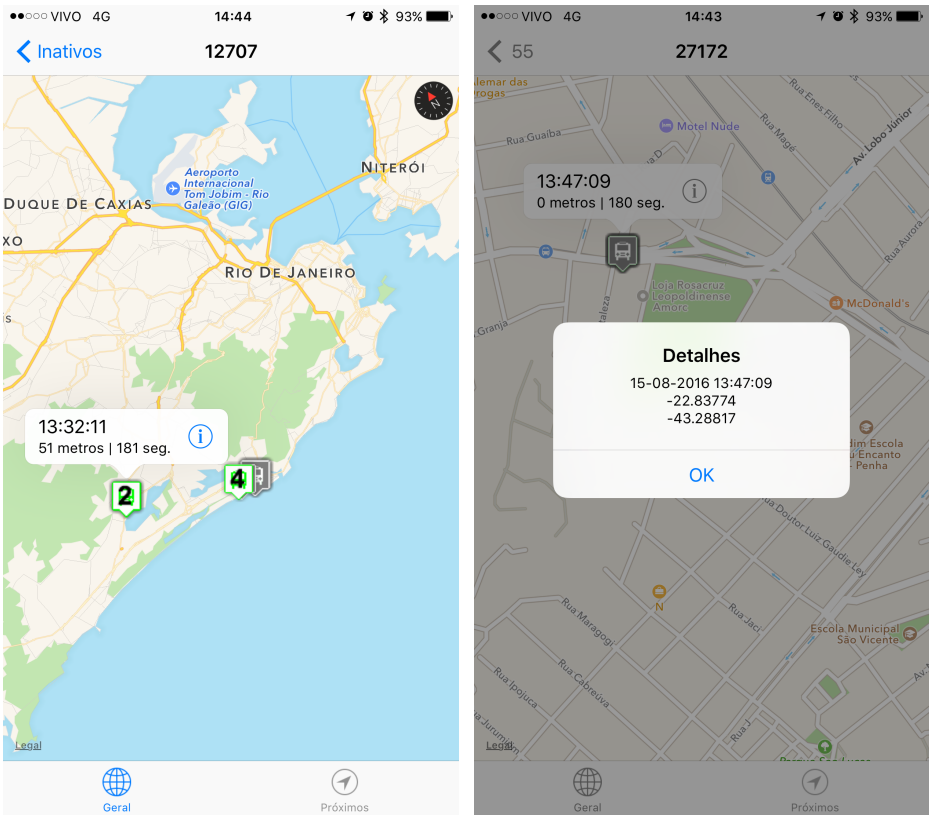


Figura 87 – Telas com detalhes do registro de um ônibus

Destacamos que esse conjunto de informações disponibilizada permite ao usuário explorar o serviço de ônibus ofertado naquele momento, amparando decisões de menor complexidade em relação ao uso dos mesmos.

6.3
Agregando Informações

As informações da Plataforma BusesInRio podem ser potencializadas com informações externas. De forma natural, como estamos utilizando um dispositivo móvel o usuário pode estar em diferentes localidades ao usar o app. Com isso, a posição atual do usuário é uma informação relevante, que pode ser utilizada para potencializar os dados da Plataforma.

A Figura 88 apresenta a tela da perspectiva de “Próximos”, acessível pela barra inferior na tela inicial do app e em outras que apresentam o menu inferior.

Nela todos os ônibus em operação cujo último registro recebido está em um raio de até 1.000 metros da sua posição atual são apresentados no mapa. O círculo azul tem sua posição como centro e simboliza o raio em questão.

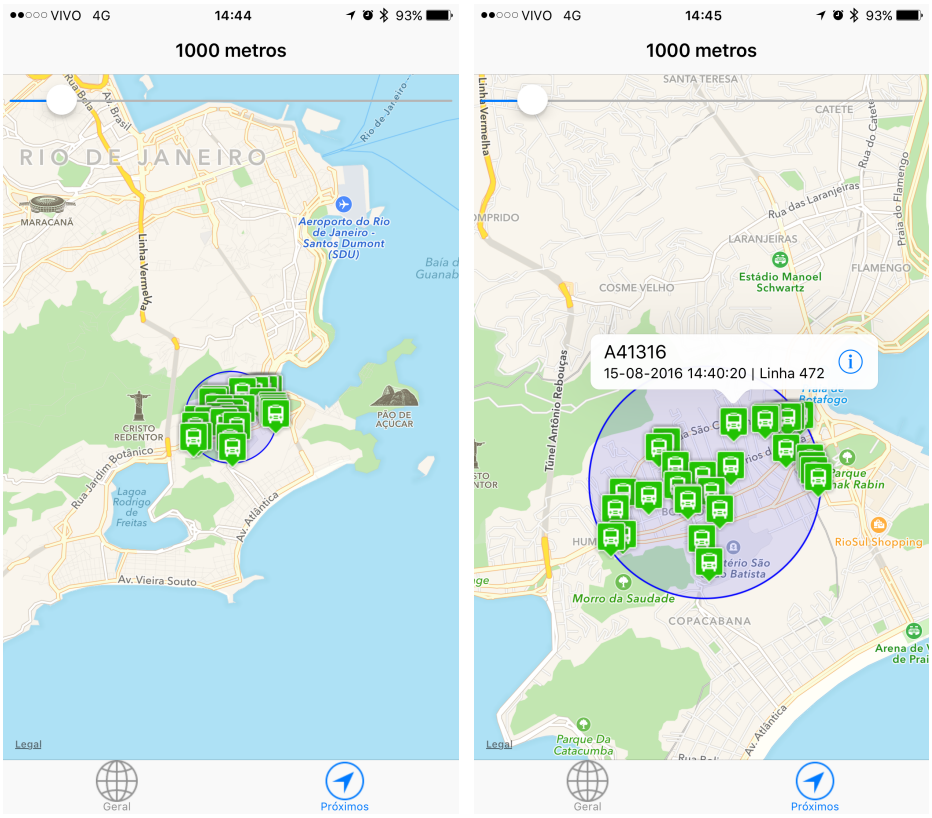


Figura 88 – Ônibus próximos da minha localidade atual

A tela apresenta ainda uma barra deslizante para que o raio, inicialmente definido como 1.000 metros, possa ser aumentado ou diminuindo. Além disso, assim como em outras telas onde os ônibus são apresentados, é possível clicar em um ônibus para obter informações adicionais, inclusive abrindo a tela para visualização dos 10 últimos registros conhecidos, a partir do ícone de informação.

Além da posição atual, podemos agregar informações de outros serviços, como, por exemplo, da Distance Matrix and Directions APIs do Google [77], que disponibiliza informações preditivas de tempo de viagem, inclusive via ônibus, que é grátis até 2.500 requisições por dia e, também, encontra-se disponível em diversos planos para volumes maiores. Seu uso é extremamente simples, bastando requisitar um endereço passando coordenadas de origem e destino e outras configurações possíveis, sendo recebido como retorno a previsão de tempo em JSON ou XML, com outros detalhes adicionais. Um exemplo de endereço é apresentado a seguir. Assim como, um exemplo de retorno na Figura 89.

https://maps.googleapis.com/maps/api/distancematrix/json?units=imperial
*&origins=**0.00**,**0.00**&destinations=**0.0**,**0.0**&key=**YOUR_API_KEY***


```
{
  "status": "OK",
  "origin_addresses": [ "" ],
  "destination_addresses": [ "" ],
  "rows": [ {
    "elements": [ {
      "status": "OK",
      "duration": {
        "value": 0,
        "text": "0 dias - horas"
      },
      "distance": {
        "value": 0,
        "text": "0 km"
      }
    },{
    "elements": [ {
      "status": "OK",
      "duration": {
        "value": 0,
        "text": "0 dias - horas"
      },
      "distance": {
        "value": 0,
        "text": "0 km"
      }
    }]
  } ]
}
```


Figura 89 – Retorno em JSON da API do Google

6.4
Informações do Serviço

As informações históricas armazenadas na Plataforma BusesInRio são por ela sumarizadas para fornecer uma visão geral do comportamento deste serviço em cada dia ao longo do tempo. Em especial, uma informação particularmente interessante é o número de ônibus e linhas distintas que enviaram dados de seus dispositivos de GPS a cada dia e, adicionalmente, quantos destes estavam

operaram nestes dias. Incorporamos no App a opção de visualizar os dados históricos em seu menu inferior, através do botão “Histórico”. A Figura 90 apresenta a tela aberta ao clicar nessa opção, que lista os dias que possuem informações sumarizadas com essas quatro informações citadas, permitindo ainda realizar filtros para localizar com maior agilidade uma data específica.

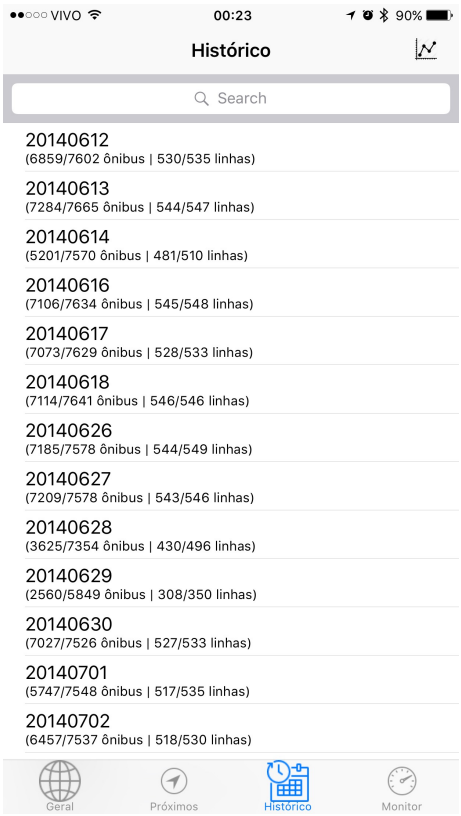


Figura 90 – Tabela com Dados Históricos no App

Para verificar se um ônibus está operando foi considerado como ponto de testagem as mudanças de hora, utilizando-se a mesma abordagem anteriormente explicada na seção 5.9, mas substituindo a hora corrente pela hora completa em questão e utilizando os dados capturados dentro da hora anteriores a essa. Consequentemente, linhas operando são as que possuem pelo menos um ônibus operando na hora cheia.

A partir desta tela é possível acessar a visão gráfica destes dados históricos, clicando no ícone de gráfico existente no canto superior direito desta tela. Esse gráfico apresenta para cada dia com dados históricos seu número de: ônibus no total; ônibus ativos; linhas de ônibus no total; linhas de ônibus ativas. Estes dados foram apresentados em gráficos na seção 5.6. Também é possível

selecionar uma data, clicando na mesma, para visualizar detalhes deste dia em específico, como, por exemplo, o número de ônibus sem linhas. Essas situações são apresentadas na Figura 91 e Figura 92.

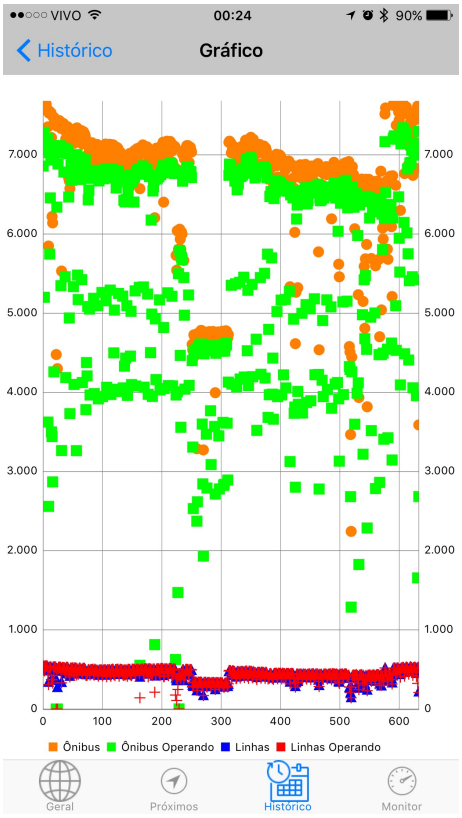


Figura 91 – Tela com Gráficos de Dados Históricos

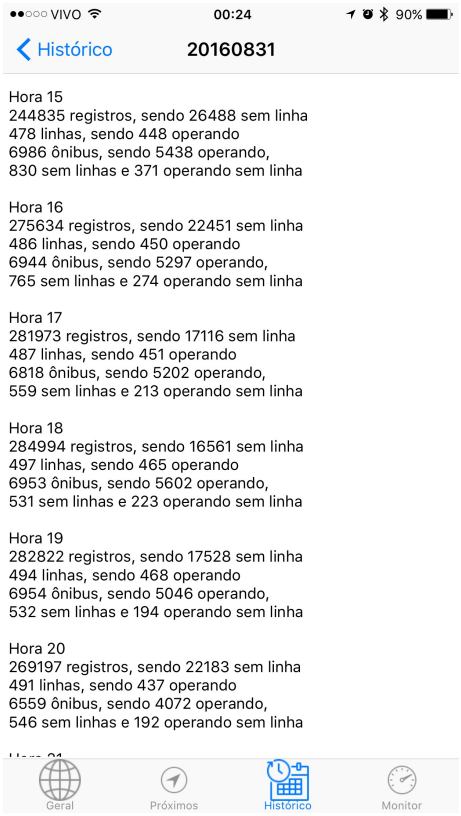


Figura 92 – Dados de Data Específica no App

6.5
Conhecendo o Tráfego

Os conhecimentos extraídos sobre os padrões do tráfego nos segmentos selecionados pode ser confrontado com a situação corrente do tráfego, conforme tratado na seção 5.6. Com base na velocidade corrente do segmento e no seu padrão, podemos novamente adotar uma escala de cores para estabelecem a situação do tráfego no segmento. Esse conhecimento é colocado à disposição do usuário no App, através da opção “Monitor” em seu menu inferior. A Figura 93 ilustra essa tela, com os segmentos destacados.

Figura 93 – Monitor do Tráfego no App

A escala de cores pode ser definida de acordo com o interesse do desenvolvedor da solução, realizando a comparação destes valores de velocidade corrente e padrão. Neste App optamos pela seguinte escala: verde para corrente 20% acima do padrão, laranja para corrente maior ou igual ao padrão, marrom para corrente até 40% do padrão e vermelho para corrente abaixo do padrão.

Nessa tela é possível visualizar esses valores de velocidade corrente padrão, assim como, o nome do segmento, simplesmente clicando sobre o mesmo, conforme ilustrado na Figura 94. Adicionalmente, clicando no botão de informação é possível conhecer o histórico deste segmento para uma data qualquer, conforme telas da Figura 95.

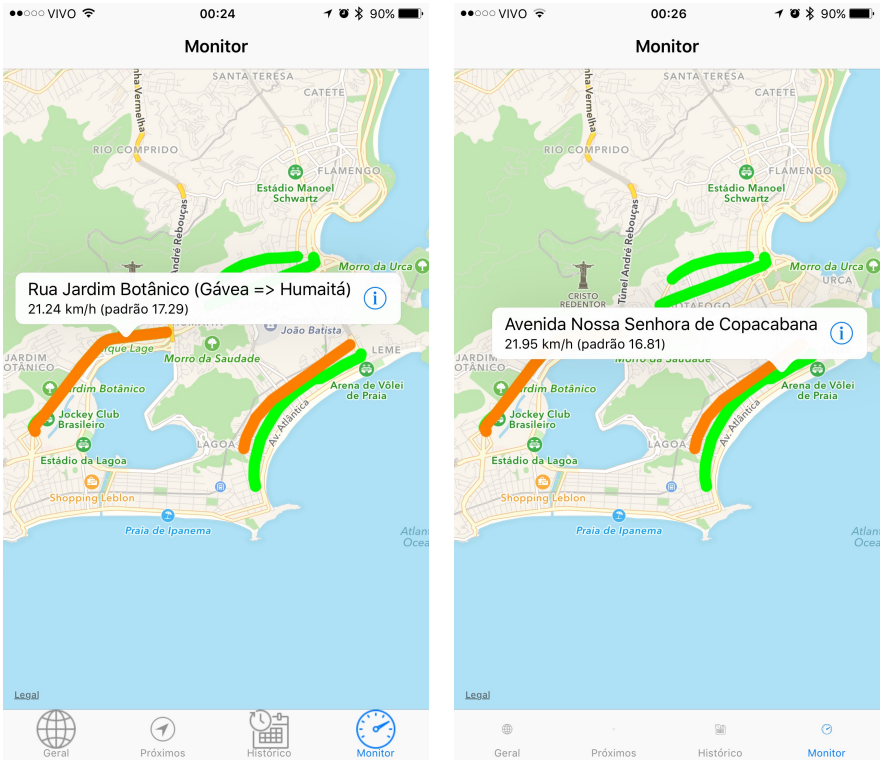


Figura 94 – Velocidade Corrente do Segmento no App

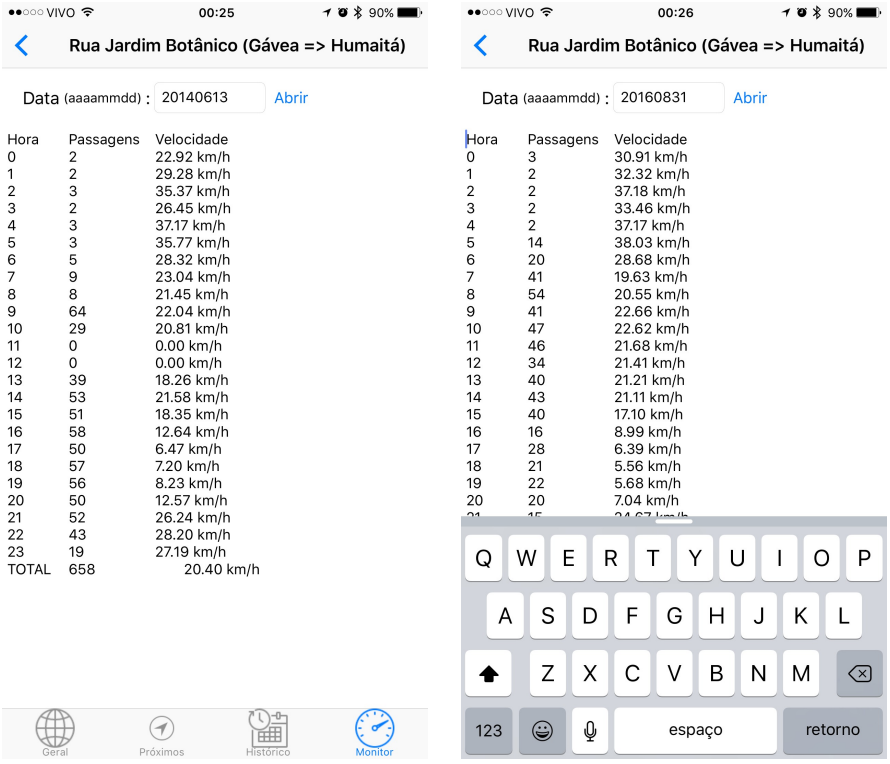


Figura 95 – Velocidade do Segmento ao Longo do Dia

Cabe registar que essas interfaces são apenas uma pequena amostra das funcionalidades

que podem ser oferecidas aos usuários a partir deste conhecimento. No entanto, essas telas já permitem apoiar o processo de decisão do usuário, conforme trataremos na seção que se segue.

6.6
Processo de Tomada de Decisão com o App

A funcionalidade de monitoramento de segmentos, apresentada na seção anterior, permite apoiar decisões do dia a dia dos moradores desta área urbana, tais como se o momento é ou não oportuno para trafegar sobre os segmentos ou se um segmento a frente apresenta fluxo mais lento, indicando que o segmento anterior tem tendência a se congestionar.

Ao mesmo tempo, a funcionalidade de visualização da situação geral permite acompanhar o deslocamento dos ônibus, verificar seus trajetos recentes e até consultar as rotas das linhas, facilitando decisões relacionadas a viagens mais imediatas, que são ainda completadas pela tela de proximidade, que permite identificar se um veículo está ou não próximo, para que o deslocamento até o local de embarque seja iniciado.

Por outro lado, a funcionalidade de visualização do histórico dos ônibus e linhas de ônibus empodera a população, que pode questionar a diminuição, manutenção ou aumento nas características do serviço (quantidade de veículos e linhas em operação) ao longo do tempo e horário.

Além disso, este App pode ser entendido como um alicerce de evoluções incrementais para os usuários destes serviço de transporte, uma vez que muitos outros conhecimentos estão disponíveis, como tratado na seção anterior, ou ainda podem ser obtidos em trabalhos futuros, em especial, no que se refere a detecção de anomalias e agregação de outras fontes de informação, com o Twitter.

7 Conclusão

7.1 Principais Contribuições e Descobertas

A Plataforma na nuvem para extração de conhecimentos a partir de dados de localização georeferenciada de mobilidade urbana, batizada de BusesInRio, é a contribuição central deste trabalho. Seu funcionamento envolve a descoberta e testagem da melhor estratégia de armazenamento e processamento deste grande volume de dados.

Além disso, ficou evidente a conceituação, aplicação e análise da visão de que os dados de GPS dos ônibus, que são transmitidos de minuto em minuto, podem representar sensores móveis de tráfego na cidade e de que as trajetórias desses ônibus podem ser entendidas como um *data stream* continuamente gerados pelos equipamentos dos ônibus.

Não obstante, algoritmos extremamente importantes foram propostos para seleção de segmentos e para extração das trajetórias desses ônibus sobre os segmentos em questão. Assim como, consolidações estatísticas foram realizadas para entender padrões e monitorar o tráfego. Sendo ainda vislumbrada a identificação de anomalias e da propagação entre segmentos de congestionamentos.

Cabe também destaque o software e app apresentados, que permitem explorar os conhecimentos extraídos de forma didática e acessível, suscitando uma infinidade de soluções que podem ser desenvolvidas consumindo a Plataforma BusesInRio.

A seguir listamos de forma estruturada as principais contribuições deste trabalho:

- Coleta e armazenamento de dados de posições de GPS (mais de 3 bilhões de entradas) de todos os ônibus que operaram na cidade desde Junho de 2014 e outras informações sobre o serviço de ônibus (pontos de parada, discretização das linhas e outras), conforme detalhado no próximo capítulo, que agora estarão disponíveis para subsidiar estudos e pesquisas futuras.
- Conceituação, aplicação e análise da visão de que os dados de GPS dos ônibus, que são transmitidos de minuto em minuto, podem representar

sensores móveis de tráfego na cidade e de que as trajetórias desses ônibus podem ser entendidas como um *data stream* continuamente gerados pelos equipamentos dos ônibus.

- Proposição, desenvolvimento e validação de uma plataforma na nuvem para extração de conhecimentos a partir de dados de localização georeferênciada de mobilidade urbana.
- Algoritmo para seleção de segmentos que representem as principais vias da cidade passíveis de monitoramento pelos sensores acima citados.
- Algoritmo para extração das trajetórias desses ônibus sobre os segmentos em questão, derivando identificadores e padrões, incluindo exemplos de análises possíveis.
- Algoritmo para monitoramento do tráfego cruzando a situação corrente com a esperada.
- Desenvolvimento de *software* desktop para visualização destes dados georeferenciados, que pode ser reutilizado para diferentes contextos.
- Desenvolvimento de aplicativo móvel para consumo dos serviços desta Plataforma, agregando características do dispositivo.

7.2 Desafios Enfrentados

Esse trabalho venceu obstáculos significativos, dentre os quais se destacam:

• Dados georeferenciados sem serem visualizados são de difícil compreensão, além disso, muitas vezes essa compreensão só é obtida ao visualizá-los e analisá-los de forma conjunta e ordenada. Inclusive, pode ser fazer necessário confrontá-los a padrões conhecidos, como o arruamento da cidade. Desenvolver um software para visualização dos dados da Plataforma BusesInRio foi fundamental para suportar o processo de criação dos algoritmos e para compreender estes dados.

• Considerando que: (i) parte relevante dos registros dos ônibus não informavam o número da linha; (ii) a maioria das linhas de ônibus não possuem rotas conhecidas; (iii) os dispositivos de GPS apresentam erros intrínsecos na ordem de 100 metros (em função das edificações, este tende a ser maior nas áreas urbanas); (iv) existem pontos cegos para os dispositivos; e (v) possivelmente

ocorrem falhas na entrega dos dados pela Prefeitura ou pelas concessionárias. Identificar a trajetória dos ônibus é uma tarefa extremamente complexa. O algoritmo implementado se pauta pela segmentação da Cidade e realiza diversas testagens para assegurar a consistência das trajetórias descobertas. Seu desenvolvimento demandou inúmeras visualizações com dados reais e aperfeiçoamentos sucessivos para alcançar resultados consistentes.

• Trabalhar com grande volume de dados exige significativo esforço computacional, tanto para seu armazenamento quanto manipulação. Desenvolver a Plataforma em um ambiente de computação em nuvem foi fundamental para alcançar um bom equilíbrio entre custo e performance.

Outros desafios enfrentados ao longo do trabalho culminaram em decisões estratégicas, das quais merecem registro:

• Nossa premissa de que um número relevante dos registros dos dispositivos de GPS dos ônibus não informavam a linha para a qual o ônibus estava servindo não se confirmou, pois de fato a situação onde esse cenário se apresenta justificava-se pelo ônibus em questão não estar operando, o que foi possível concluir com a extração deste conhecimento a partir dos dados.

• O intervalo com que os dados são coletados não permitiu a definição de uma estratégia conclusiva para identificação de pontos de parada, assim como, os pontos de parada informados pela Prefeitura se mostraram inconsistentes. Dessa forma, o estabelecimento de tabelas de horários para pontos de parada se mostrou um desafio inatingível a partir destes dados. As tabelas de horário podem ser pensadas a partir da lógica de segmentos, porém esse foco não foi incluído nesse trabalho, uma vez que a abrangência da Plataforma BusesInRio ficou evidenciada pelos demais conhecimentos extraídos.

• A estratégia adotada a partir da segmentação da cidade permite estimar o tempo de deslocamento dentro destes segmentos com boa precisão, no entanto para cálculo do tempo estimado de chegada seria necessário calcular todos os segmentos presentes na linha. Como no período de desenvolvimento deste trabalho o Google investiu fortemente nesta funcionalidade para atendimento aos Jogos Olímpicos, o foco neste serviço foi retirado dos objetivos do trabalho. Inclusive, o Google disponibiliza uma API para acesso a esses dados.

7.3
Lições Aprendidas

A principal lição aprendida neste trabalho é que ao desenvolvermos soluções que transitam entre o mundo real (ônibus na rua) e virtual (resultados dos algoritmos de extração de conhecimento da Plataforma BusesInRio), em um cenário de volumes consideráveis e diariamente crescente de dados geográficos, é indispensável visualizar dados reais antes e depois da aplicação dos algoritmos repetidas vezes, para que se compreenda os efeitos reais do que se está na teoria buscando.

Tomando como exemplo a identificação das trajetórias dos ônibus sobre os segmentos, foi necessário repensar diversas vezes algoritmos que teoricamente estavam corretos, mas que na prática, não entregaram um resultado satisfatório. Quem imaginaria de antemão que um GPS poderia ficar sem enviar dados durante horas e que o último dado recebido e o primeiro, quando retornou a transmissão de dados do GPS, eram respectivamente próximos ao início e fim de um segmento, criando uma falsa impressão de uma trajetória correta, mas na prática totalmente equivocada para o propósito de monitorar um segmento.

Portanto, em síntese, é preciso experimentar com dados reais qualquer proposta teórica, preferencialmente em exaustão.

7.4
Limitações

A Plataforma BusesInRio basea-se em serviços de tecnologia proprietária da Microsoft, o Windows Azure, consequentemente, a capacidade de processamento e armazenamento está diretamente vinculada aos recursos financeiros disponíveis, no entanto, apresenta-se uma relação otimizada de custo-benefício para a solução.

Outro ponto que merece destaque, é que o foco deste trabalho não foi apurar algoritmos de predição (e.g. a velocidade média esperada em determinado horário para um segmento em específico), portanto, uma testagem de campo e refinamentos estatísticos são oportunos para os que pretendem usufruir destes benefícios.

Por último, cabe registro que o cálculo de tempo estimado de chegada, tabela de horários e identificação de pontos de parada, foram assuntos inicialmente vislumbrados, que perderam prioridade em função do foco na

proposição de uma plataforma sólida para realização de Data Science, mas que podem ser endereçados em trabalhos futuros.

7.5 Trabalhos Futuros

Os resultados alcançados neste trabalho motivam a realização de trabalhos futuros, a saber:

- A estratégia implementada de segmentação da área urbana tem a característica de ser gulosa, ou seja, basea-se na repetição extensiva do procedimento para todas as vias da cidade, no entanto, algoritmos mais refinados poderiam ser propostos para seleção destes segmentos, assim como, estratégias para particionar esses segmentos em segmentos menores poderiam ser elaboradas, conforme abordado em [76].
- A detecção de trajetórias implementada demonstrou grande potencial para detecção de anomalias no tráfego e estudo da propagação de anomalias pelos segmentos conectados. Portanto, um estudo mais aprofundado pode ser conduzido, inclusive relacionando outras fontes de dados (e.g. Twitter), para implementação de novos serviços na Plataforma e o acesso dos mesmos pelo App para população das áreas urbanas.
- O conhecimento extraído a respeito dos segmentos possibilita um profundo estudo estatístico destas vias, que certamente podem amparar um melhor planejamento da mobilidade urbana, ou mesmo, questionamentos da sociedade a seus governantes. Assim como, a análise de situações especificas (ex. aumento do túnel X melhorou o tráfego na via Y). Esse estudo também permitiria a disponibilização de um site (busesinrio.com) com indicadores públicos do serviço de ônibus municipal da Cidade do Rio de Janeiro.
- Utilização da plataforma para desenvolvimento de algoritmos de tempo estimado de chegada, descoberta de tabela de horários e identificação de pontos de parada das linhas de ônibus.
- Por último, uma proposta interessante seria o uso da Plataforma BusesInRio com dados de outra cidade, por exemplo, de São Paulo, e de outros modais, por exemplo, Taxis.

Referências bibliográficas

[1] SHI, WENHUAN, QING-JIE KONG, AND YUNCAI LIU. "A GPS/GIS integrated system for urban traffic flow analysis." 2008 *11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2008.

[2] ZENG, WEI, CHI-WING FU, STEFAN MÜLLER ARISONA, ALEXANDER ERATH, AND HUAMIN QU. "Visualizing mobility of public transportation system." *IEEE transactions on visualization and computer graphics 20*, no. 12 (2014): 1833-1842.

[3] ZHANG, CHA, AND YUNQIAN MA. "Ensemble machine learning." Springer, 2012.

[4] ZHANG, LIANGPEI, LEFEI ZHANG, AND BO DU. "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art." *IEEE Geoscience and Remote Sensing Magazine 4.2* (2016): 22-40.

[5] ZHENG, Y., CAPRA, L., WOLFSON, O., YANG, H. *Urban computing. ACM Trans. Intell. Syst.Technol. 5* (2014), 1–55.

[6] DADOS RIO. Prefeitura da Cidade do Rio de Janeiro. Disponível em: <http://data.rio/ >. Acessado em 14/09/2016.

[7] DHAR, VASANT. "Data science and prediction". *Communications of the ACM 56.12* (2013): 64-73.

[8] CAO, LONGBING. "Data science and analytics: a new era." *International Journal of Data Science and Analytics* 1.1 (2016): 1-2.

[9] The Economist (2015), Artificial Intelligence, "Rise of the machines". Disponível em <http://www.economist.com/news/briefing/21650526-artificial-intelligence-scares-peopleexcessively-so-rise-machines>. Acessado em 14/09/2016.

[10] ASAMOAH, DANIEL, DEREK DORAN, AND SHU SCHILLER. "Teaching the Foundations of Data Science: An Interdisciplinary Approach." *arXiv preprint arXiv:1512.04456* (2015).

[11] WALLER, MATTHEW A., AND STANLEY E. FAWCETT. "Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management."*Journal of Business Logistics* 34.2 (2013): 77-84.

[12] GARTNER <www.gartner.com>. Acessado em 14/09/2016.

[13] GUBERFAIN, B.; CÔRTES VIEIRA, H. "A visual analysis of bus GPS data in Rio". Pontifícia Universidade Católica do Rio de Janeiro, 2015.

[14] BARBOSA, L.; KORMÁKSSON, M.; VIEIRA, M. R.; TAVARES, R. L.; ZADROZNY, B. "Vistradas: Visual Analytics for Urban Trajectory Data." *GEOINFO. XV Brazilian Symposium on GeoInformatics*.

[15] PU, J., LIU, S., DING, Y., QU, H., AND NI, L. M. (2013). "T-watcher: A new visual analytic system for effective traffic surveillance." *In Proc. of IEEE MDM, pages 127–136*.

[16] LU, C.-T., BOEDIHARDJO, A. P., AND ZHENG, J. (2006). "Aitvs: Advanced interactive traffic visualization system*." In Proc. of IEEE ICDE, pages 167–167*.

[17] SHEKHAR, S., LU, C. T., LIU, R., AND ZHOU, C. (2002). "Cubeview: a system for traffic data visualization." *In Proc. of IEEE Int'l. Transp. Sys., pages 674–678*.

[18] GEISLER, S.; QUIX, C.; SCHIFFER, S.; JARKE, M. "An evaluation framework for traffic information systems based on data streams." *Transportation Research Part C: Emerging Technologies*, v. 23, p. 29–55, 2012.

[19] YAN, Z., SPREMIC, L., CHAKRABORTY, D., PARENT, C., SPACCAPIETRA, S., AND ABERER, K. (2010). "Automatic construction and multi-level visualization of semantic trajectories." In Proc. of ACM SIGSPATIAL, pages 524–525.

[20] ZHANG, J.-D.; XU, J.; LIAO, S. S. "Aggregating and sampling methods for processing GPS data streams for traffic state estimation." *IEEE Transactions on Intelligent Transportation Systems*, v. 14, n. 4, p. 1629–1641, 2013.

[21] ALBUQUERQUE, F.C., "Environment changes detection: A proactive system to monitor moving objects." Dissertação de Mestrado, PUC-RJ, 2012.

[22] ZHU, B.; XU, X. "Urban Principal Traffic Flow Analysis Based on Taxi Trajectories Mining. *International Conference in Swarm Intelligence.* Anais, 2015.

[23] ALBUQUERQUE, F.C., M.C. CASANOVA, H. LOPES, L.R. REDLICH, J.A.F. MACEDO, M. LEMOS, M.T.M. CARVALHO, C. RENSO. "A methodology for traffic-related Twitter messages interpretation." In Computers in Industry, Vol. 78, pages 57 - 69, Elsevier, 2016.

[24] REDLICH, L.R. "Modelagem de eventos de trânsito com base em clipping de grandes massas de dados da Web." Dissertação de Mestrado, PUC-RJ, 2013.

[25] KUMAR, P.; RANGANATH, S.; WEIMIN, H.; SENGUPTA, K. "Framework for real-time behavior interpretation from traffic video*." IEEE Transactions on Intelligent Transportation Systems*, v. 6, n. 1, p. 43–53, 2005.

[26] SUN, D.; LUO, H.; FU, L.; *et al.* "Predicting bus arrival time on the basis of global positioning system data." *Transportation Research Record: Journal of the Transportation Research Board*, n. 2034, p. 62–72, 2007.

[27] GUNJAL SUNIL, N.; JOSHI AJINKYA, V.; GOSAVI SWAPNIL, C.; KSHIRSAGAR VYANKTESH, B. Dynamic Bus Timetable Using GPS. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol, v. 3, n. 3, p. 775–778, 2014.

[28] HOU, L.; ZHANG, Z.; LU, B.; XU, R.; ZHANG, Y. " Estimation of incident-induced congestion on signalized arteries using traffic sensor data." *Tsinghua Science and Technology*, 2012.

[29] CHEN, C.; ZHANG, D.; CASTRO, P. S.; *et al.* "Real-time detection of anomalous taxi trajectories from GPS traces." *International Conference on*

Mobile and Ubiquitous Systems: Computing, Networking, and Services. Anais, 2011.

[30] ARANA, P.; CABEZUDO, S.; PEÑALBA, M. Influence of weather conditions on transit ridership: A statistical study using data from Smartcards. Transportation Research Part A: Policy and Practice, v. 59, p. 1–12, jan 2014.

[31] KHATTAK, A.; WANG, X.; ZHANG, H. "Incident management integration tool: dynamically predicting incident durations, secondary incident occurrence and incident delays." *IET Intelligent Transport Systems*, v. 6, n. 2, p. 204–214, 2012.

[32] CHUNG, Y.-S.; CHIOU, Y.-C.; LIN, C.-H. "Simultaneous equation modeling of freeway accident duration and lanes blocked." *Analytic Methods in Accident Research*, v. 7, p. 16–28, 2015.

[33] KUANG, W.; AN, S.; JIANG, H. "Detecting traffic anomalies in urban areas using taxi GPS data." *Mathematical Problems in Engineering*, v. 2015, 2015.

[34] KOETSE, M. J.; RIETVELD, P. "The impact of climate change and weather on transport: An overview of empirical findings." *Transportation Research Part D: Transport and Environment*, v. 14, n. 3, p. 205–221, 2009.

[35] LI, R. "Traffic incident duration analysis and prediction models based on the survival analysis approach." *IET Intelligent Transport Systems*, v. 9, n. 4, p. 351–358, 2015.

[36] MILLER, M.; GUPTA, C. "Mining traffic incidents to forecast impact." *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*.

[37] PAN, B.; DEMIRYUREK, U.; GUPTA, C.; SHAHABI, C. "Forecasting spatiotemporal impact of traffic incidents for next-generation navigation systems." *Knowledge and Information Systems*, v. 45, n. 1, p. 75–104, 2014.

[38] TAVASSOLI HOJATI, A. "Modelling the impact of traffic incidents on travel time reliability." *The University of Queensland*, 2014.

[39] WANG, J.; LIU, B. "Modeling the duration and queue length of Urban Expressway incident." *2015 International Conference on Transportation Information and Safety (ICTIS)*. *IEEE*.

[40] XIE, W.; WANG, J. "Modelling the impact scope of urban express way incident." *2015 International Conference on Transportation Information and Safety (ICTIS)*. *IEEE*.

[41] ESTER, M., KRIEGEL, H., SANDER, J., XU, X.. "A density-based algorithm for discovering clusters in large spatial databases with noise." *In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining,* (1996): 226–231.

[42] ESTER, M., KRIEGEL, H.-P., SANDER, J., WIMMER, M., XU, X. "Incremental clustering for mining in a data warehousing environment." *In: Proceedings of the 24th International Conference on Very Large Data Bases*, San Francisco, CA, USA, (1998): 323–333.

[43] FRED, ANA LN, AND ANIL K. JAIN. "Combining multiple clusterings using evidence accumulation." *IEEE transactions on pattern analysis and machine intelligence* 27.6 (2005): 835-850.

[44] RORIZ JUNIOR, M., M. ENDLER, F. SILVA E SILVA, M. A. CASANOVA AND H. LOPES. "A heuristic approach for on-line discovery of unidentified spatial clusters from grid-based streaming algorithms". *8th International Conference on Big Data Analytics and Knowledge Discovery* - DaWaK (2016).

[45] AMINI, A., WAH, T., SABOOHI, H. "On density-based data streams clustering algorithms: a survey." *J. Comput. Sci. Technol. 29* (2014), 116–141.

[46] JOSE, D.; PRASAD, S.; SRIDHAR, V. G. "Intelligent vehicle monitoring using global positioning system and cloud computing." *Procedia Computer Science*, v. 50, p. 440–446, 2015.

[47] ARBOLEDA, M. A.; PARRA, I. F.; ARISTIZÁBAL, I.; SABOGAL, H. "Estudio dinámico de la movilidad en la ciudad de Santiago de Cali - Colombia. X Congreso" *Latinoamericano de Dinámica de Sistemas.* Buenos Aires, Argentina.

[48] KARGUPTA, H.; SARKAR, K.; GILLIGAN, M. "MineFleet®: an overview of a widely adopted distributed vehicle performance data mining system. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* 2010.

[49] IBGE. Acessado em 14/09/2016. Disponível em: <http://cidades.ibge.gov.br/xtras/temas.php?lang=&codmun=330455&idtema=16&search=||s%EDntese-das-informa%E7%F5es>.

[50] IPEA. Acessado em 14/09/2016. Disponível em: <http://www.ipea.gov.br/portal/images/stories/PDFs/SIPS/110124_sips_mobilidade.pdf>.

[51] Constituição Federal de 1998. Acessada em 14/09/2016. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/constitui%C3%A7ao.htm>.

[52] Lei Federal 12.581. Acessada em 14/09/2016. Disponível em <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12587.htm>.

[53] DAPP. Acessado em 14/09/2016. Disponível em: <http://dapp.fgv.br/sites/default/files/document.pdf>.

[54] DATABANK FETRANSPOR/RIOÔNIBUS de 13/04/2015, sendo 2003 e 2012 do PDU e 2016 projeções. Disponível em FETRANSPOR.

[55] FETRANSPOR. Acessado em 14/09/2016. Disponível em: <https://www.fetranspor.com.br/mobilidade-urbana-setor-em-numeros>.

[56] IBGE. Acessado em 14/09/2016. Disponível em: <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/sinopse.pdf>.

[57] GTFS. Acessado em 14/09/2016. Disponível em <https://developers.google.com/transit/gtfs/>.

[58] Microsofr Azure. Disponível em <https://azure.microsoft.com/en-us/>. Acessado 31/08/20-16.

[59] RAMSEY, P. PostGIS manual. Disponível em <http://postgis.refractions.net/documentation>. Acessado em 14/09/2016.

[60] MARCO ANTONIO CASANOVA, GILBERTO CÂMARA, CLODOVEU A. DAVIS JR., LÚBIA VINHAS E GILBERTO RIBEIRO DE QUEIROZ. " Bancos de Dados Geográficos". Disponível em <http://www.inf.puc-rio.br/~casanova/Publications/Books/2005-BDG.pdf>. Acessado em 14/09/2016.

[61] CUNHA, T. M. DE A. "Escalabilidade de Sistemas com Banco de Dados NoSQL: um Estudo de Caso Comparativo com MongoDB e MySQL. 2011." *85 f. Trabalho de Conclusão de Curso* (Ciência da Computação) – Centro Universitário da Bahia – Estácio, Salvador.

[62] MONGODB. Disponível em <https://www.mongodb.org/>. Acesso em 14/09/2016.

[63] S. SCHMID, ESZTER GALICZ, WOLFGANG REINHARDT. Performance investigation of selected SQL and NoSQL databases." *AGILE 2015 – Lisbon*, June 9-12, 2015. University of the Bundeswehr.

[64] "Performance and scale testing with Azure DocumentDB". Disponível em <https://azure.microsoft.com/en-us/documentation/articles/documentdb-performance-testing/>. Acessado em 14/09/2016.

[65] Manual Dados Abertos do Governo Federal. Acessado em 14/09/2016. Disponível em <http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual_Dados_Abertos_WEB.pdf>.

[66] W. GELLERT, S. GOTTWALD, M. HELLWICH, H. KÄSTNER, AND H. KÜSTNER. *"The VNR Concise Encyclopedia of Mathematics"*, 2nd ed., ch. 12 (Van Nostrand Reinhold: New York, 1989).

[67] N. ZUMEL AND J. MOUNT. "Practical Data Science with R."*Manning Publications*, 2014.

[68] V. MAYER-SCHONBERGER AND K. CUKIER. "Big data: A revolution that will transform how we live, work, and think." *Houghton Mifflin Harcourt*, 2013.

[69] Z. KHAN, A. ANJUM, K. SOOMRO, AND T. MUHAMMAD. "Towards cloud based big data analytics for smart future cities*," Journal of Cloud Computing: Advances, Systems and Applications*, 2015.

[70] "Cartilha Acesso a Informação." Acessada em 14/09/2016. Disponível em <http://www.acessoainformacao.gov.br/central-de-conteudo/publicacoes/arquivos/cartilhaacessoainformacao.pdf>.

[71] "Definição de Dados Abertos". Disponível em <http://opendefinition.org/>. Acessado em 14/09/016.

[72] CÂMARA, G. "Representação computacional de dados geográficos."In: CASANOVA.

[73] "IT INDUSTRY OUTLOOK 2016". Disponível em <https://www.comptia.org/resources/it-industry-outlook-2016-final>. Acessado em 14/09/2016.

[74] FROZZA, A. A. "Plataforma UrbanMob: Infraestrutura para armazenamento de trajetórias urbanas de objetos móveis." *Projeto de IC submetido ao edital* IFC 037/GDG/IFC-CAM/2012. Camboriú, 2012.

[75] AMARAL, BRUNO GUBERFAIN DO, RAFAEL NASSER, MARCO ANTONIO CASANOVA, AND HÉLIO LOPES. "BusesinRio: Buses as Mobile Traffic Sensors: Managing the Bus GPS Data in the City of Rio de Janeiro." 2016. *17th IEEE International Conference on Mobile Data Management* (MDM). Vol. 1. IEEE, 2016.

[76] RODRIGUEZ, KATHRIN, MARCO A. CASANOVA, LUIZ ANDRÉ PAES LEME, HÉLIO LOPES, RAFAEL NASSER, AND BRUNO GUBERFAIN DO AMARAL. "On the Design of a Traffic Observatory Application Based on Bus Trajectories." *18th International conference on Enterprise Information Systems* (2016).

[77] Google Maps API – Distance Matrix. Acessado em 14/09/2016. Disponível em <https://developers.google.com/maps/documentation/distance-matrix/>.

[78] GMap.Net do MIT. Acessível em <https://greatmaps.codeplex.com/>. Acessado em 14/09/2016.

[79] RAFAEL NASSER. McCloud Service Framework: Arcabouço para desenvolvimento de serviços baseados em Simulação de Monte Carlo na Cloud. Defesa de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro. 2012.

[80] OpenStreetMap. Disponível em <http://www.openstreetmap.org/>. Acessado em 14/09/2016.

[81] Google Maps. Disponível em <https://www.google.com.br>. Acessado em 14/09/2016.

[82] QGIS. Disponível em <www.qgis.org>. Acessado em 14/09/2016.

[83] SOAEngine. Disponível em <https://github.com/priore/SOAPEngine>. Acessado em 14/092016.

[84] M-Atlas. Disponível em < http://m-atlas.eu/>. Acessao em 21/10/2016.

[85] WCF em MSDN. Disponível em <https://msdn.microsoft.com/pt-br/library/ms731082(v=vs.110).aspx>. Acessado em 21/10/2016.